

Approaching Throughput-optimality in Distributed CSMA Scheduling Algorithms with Collisions

Libin Jiang and Jean Walrand

EECS Department, University of California at Berkeley
{ljjiang,wlr}@eecs.berkeley.edu

Abstract—It was shown recently that CSMA (Carrier Sense Multiple Access)-like distributed algorithms can achieve the maximal throughput in wireless networks (and task processing networks) under certain assumptions. One important, but idealized assumption is that the sensing time is negligible, so that there is no collision. In this paper, we study more practical CSMA-based scheduling algorithms with collisions. First, we provide a Markov chain model and give an explicit throughput formula which takes into account the cost of collisions and overhead. The formula has a simple form since the Markov chain is “almost” time-reversible. Second, we propose transmission-length control algorithms to approach throughput optimality in this case. Sufficient conditions are given to ensure the convergence and stability of the proposed algorithms. Finally, we characterize the relationship between the CSMA parameters (such as the maximum packet lengths) and the achievable capacity region.

Index Terms—Distributed scheduling, CSMA, Markov chain, convex optimization

I. INTRODUCTION

Efficient resource allocation is essential to achieve high utilization of a class of networks with resource-sharing constraints, such as wireless networks and stochastic processing networks (SPN [5]). In wireless networks, certain links cannot transmit at the same time due to the interference constraints among them. In a SPN, two tasks cannot be processed simultaneously if they both require monopolizing a common resource. A scheduling algorithm determines which link to activate (or which task to process) at a given time without violating these constraints. Designing efficient distributed scheduling algorithms to achieve high throughput is especially a challenging task [4].

Maximal-weight scheduling (MWS) [3] is a classical *throughput-optimal* algorithm. That is, MWS can stabilize all queues in the network as long as the arrival rates are within the capacity region. MWS operates in slotted time. In each slot, a set of non-conflicting links (called an “independent set”, or “IS”) that have the maximal weight (i.e., summation of queue lengths) are scheduled. However, implementing MWS in general networks is quite difficult for two reasons. (i) MWS is inherently a centralized algorithm and is not amenable to distributed implementation; (ii) finding a maximal-weighted IS

(in each slot) is NP-complete in general and is hard even for centralized algorithms.

On the other hand, there has been active research on low-complexity but suboptimal scheduling algorithms. For example, reference [6] shows that Maximal Scheduling can only guarantee a fraction of the network capacity. A related algorithm has been studied in [7] in the context of IEEE 802.11 networks. Longest-Queue-First (LQF) algorithm (see, for example, [9], [10], [11], [12]), which greedily schedules queues in the descending order of the queue lengths, tends to achieve higher throughput than Maximal Scheduling, although it is not throughput-optimal in general [10]. Reference [13] proposed random-access-based algorithms that can achieve performance comparable to that of maximal-size scheduling.

Recently, we proposed a distributed adaptive CSMA (Carrier Sense Multiple Access) algorithm [16] that is throughput-optimal for a general interference model, under certain assumptions (further explained later). The algorithm has a few desirable features. It is distributed (i.e., each node only uses its own backlog information), asynchronous (i.e., nodes do not need to synchronize their transmission) and requires no control message. (In [17], Rajagopalan and Shah independently proposed a similar randomized algorithm in the context of optical networks. Reference [18] showed that under a “node-exclusive” interference model, CSMA can be made throughput-optimal in an asymptotic regime.) We have also developed a joint algorithm in [16] that combined the adaptive CSMA scheduling with congestion control to approach the maximal total utility of competing data flows.

However, the algorithms in [16] assume an idealized CSMA protocol ([14], [25], [26], [27]), meaning that the sensing is instantaneous, so that conflicting links do not transmit at the same time (i.e., collisions do not occur). In many situations, however, this is an unnatural assumption. For example, in CSMA/CA wireless networks, due to the propagation delay and processing time, sensing is not instantaneous. Instead, time can be viewed as divided into discrete minislots, and collisions happen if multiple conflicting links try to transmit in the same minislot. When a collision occurs, all links that are involved lose their packets, and will try again later.

In this paper, we study this important practical issue when designing CSMA-based scheduling algorithms. We follow four main steps: (1) we first present a Markov chain model for a simple CSMA protocol with collisions, and give an explicit throughput formula (in section II) that has a simple form

This work is supported by MURI grant BAA 07-036.18.

Some main results of this paper appeared in [29].

This is the longer version of a same-titled paper to appear in *IEEE/ACM Transactions on Networking*.

since the Markov chain is “almost” time-reversible; (2) we show that the algorithms in [16] can be extended to approach throughput optimality even with collisions (section III); (3) we give sufficient conditions to ensure the convergence/stability of the proposed algorithms (section III); (4) finally, we discuss the tradeoff between the achievable capacity region and short-term fairness, and we characterize the relationship between the CSMA parameters (such as the maximum packet lengths) and the achievable capacity region.

Although step (2) can be viewed as a generalization of [16], we believe that this generalization is important and non-trivial. The importance, as mentioned above, is because collisions are unavoidable in CSMA/CA wireless networks, and the collision-free model used in [16] does not provide enough accuracy. The generalization is non-trivial for the following reasons. Step (2) requires the result of step (1). In step (1), the Markov chain used to model the CSMA protocol is no longer time-reversible as in [16]. We need to re-define the state space in order to compute the service rates it can provide. Interestingly, as a result of our design the chain is “almost” time-reversible which can be exploited. In step (2), in view of the expression of service rates derived in step (1), it is important to realize that adjusting the backoff times as in [16] does not lead to the desirable throughput-optimal property. Instead, one should adjust the transmission lengths. Different from [16], we further show that the “optimal” CSMA parameters (in this case the mean transmission lengths) are *unique*. This fact is needed to establish the convergence of our algorithm in step (3).

We note that in a recent work [8], Ni and Srikant also proposed a CSMA-like algorithm to achieve near-optimal throughput with collisions taken into account. The algorithm in [8] uses synchronized and alternate control phase and data phase. It is designed to realize a discrete-time CSMA (in the data phase) which has the same stationary distribution as its continuous counterpart in [16]. The control phase does not contribute to the throughput and can be viewed as the protocol overhead. Different from [8], our algorithm here is asynchronous, and has more resemblance to the RTS/CTS mode in IEEE 802.11. Although the algorithm in [8] is quite elegant and could potentially be applied to other time-slotted systems, we believe that it is an interesting problem to understand how to use the asynchronous algorithm to achieve throughput-optimality. Also, due to its similarity to the RTS/CTS mode of IEEE 802.11, the throughput analysis in this paper could also deepen the understanding of 802.11 in general topologies.

II. CSMA/CA-BASED SCHEDULING WITH COLLISIONS

A. Model

In this section we present a model for CSMA/CA-based scheduling with collisions. Note that the goal of the paper is *not* to propose a comprehensive model that captures all details for IEEE 802.11 networks and predict the performance of such networks (The literature in that area has been very rich. See, for example, [24], [25] and the references therein.) Instead, at a

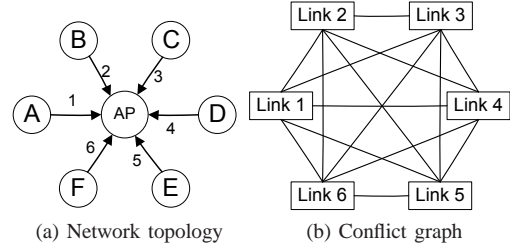


Fig. 1: Infrastructure network

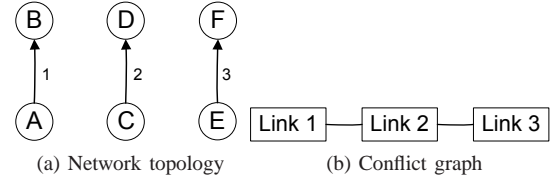


Fig. 2: Ad-hoc network

more abstract level, we are interested in a distributed scheduling algorithm that is inspired by CSMA/CA, and designing adaptive algorithms to approach throughput-optimality.

Consider a (single-channel) wireless network. Define a “link” as an (ordered) transmitter-receiver pair. Assume that there are K links, and denote the set of links by \mathcal{N} (then, $K = |\mathcal{N}|$). Without loss of generality, assume that each link has a capacity of 1. We say that two links *conflict* if they cannot transmit (or, “be active”) at the same time due to interference. (The conflict relationship is assumed to be symmetric.) Accordingly, define G as the conflict graph. Each vertex in G represents a link, and there is an edge between two vertexes if the corresponding links conflict. Note that this simple conflict model may not reflect all possible interference structures that could occur in wireless networks. However, it does provide a useful abstraction and has been used widely in literature (see, for example, [25] and [4]).

Fig. 1 (a) shows a wireless LAN with 6 links. The network’s conflict graph is a full graph (Fig. 1 (b)). (Circles represent nodes and rectangles represent links.) Fig. 2 (a) shows an ad-hoc network with 3 links. Assume that link 1, 2 conflict, and link 2, 3 conflict. Then the network’s conflict graph is Fig. 2 (b).

Basic Protocol

We now describe the basic CSMA/CA protocol with fixed transmission probabilities (which suffices for our later development.) Let $\tilde{\sigma}$ be the duration of each idle slot (or “minislot”). ($\tilde{\sigma}$ should be at least the time needed by any wireless station to detect the transmission of any other station. Specifically, it accounts for the propagation delay, the time needed to switch from the receiving to the transmitting state, and the time to signal to the MAC layer the state of the channel [15]. The value of $\tilde{\sigma}$ varies for different physical layers. In IEEE 802.11a, for example, $\tilde{\sigma} = 9\mu s$.) In the following we will simply use “slot” to refer to the minislot.

Assume that all links are saturated (i.e., always have packets to transmit). In each slot, if (the transmitter of) link i is not

already transmitting and if the medium is idle, the transmitter of link i starts transmission with probability p_i . If at a certain slot, link i did not choose to transmit but a conflicting link starts transmitting, then link i keeps silent until that transmission ends. If conflicting links start transmitting at the same slot, then a collision happens. [We assume that the network has no hidden node (HN). The case with HNs will be discussed in Section V-C. For possible ways to address the HN problem, please refer to [19] and its references.]

Each link transmits a short probe packet with length γ (similar to the RTS packet in 802.11) before the data is transmitted. (All “lengths” here are measured in number of slots and are assumed to be integers.) This increases the overhead of successful transmissions, but can avoid collisions of long data packets. When a collision happens, only the probe packets collide, so each collision lasts a length of γ . Assume that a successful transmission of link i lasts τ_i , which includes a constant overhead τ' (composed of RTS, CTS, ACK, etc) and the data payload τ_i^p which is a random variable. Clearly $\tau_i \geq \tau'$. Let the p.m.f. (probability mass function) of τ_i be

$$Pr\{\tau_i = b_i\} = P_i(b_i), \forall b_i \in \mathcal{Z}_{++} \quad (1)$$

and assume that the p.m.f. has a *finite support*, i.e., $P_i(b_i) = 0, \forall b_i > b_{max} > 0$.¹ Then the mean of τ_i is

$$T_i := E(\tau_i) = \sum_{b \in \mathcal{Z}_{++}} b \cdot P_i(b). \quad (2)$$

Fig. 3 illustrates the timeline of the 3-link network in Fig. 2 (b), where link 1 and 2 conflict, and link 2 and 3 conflict.

We note a subtle point in our modeling. In IEEE 802.11, a link can attempt to start a transmission only after it has sensed the medium as idle for a constant time (which is called DIFS, or “DCF Inter Frame Space”). To take this into account, DIFS is included in the packet transmission length τ_i and the collision length γ . In particular, for a successful transmission of link i , DIFS is included in the constant overhead τ' . Although DIFS, as part of τ' , is actually after the payload, in Fig. 3 we plot τ' before the payload. This is for convenience and does not affect our results. So, under this model, a link can attempt to start a transmission *immediately* after the transmissions of its conflicting links end.

The above model is almost time-reversible such that a simple throughput formula can be derived. A process is “time-reversible” if the process and its time-reversed process are statistically indistinguishable [1]. Our model, in Fig. 3, reversed in time, follows the same protocol as described above, except for the order of the overhead and the payload, which are reversed. A key reason for this property is that the collisions start and finish at the same time. (This point will be made more precise in Appendix A.)

B. Notation

Let the “on-off state” be $x \in \{0, 1\}^K$ where x_k , the k -th element of x , is such that $x_k = 1$ if link k is active

¹The finite support assumption is not a restrictive one, since in practical wireless networks there is usually an upper bound on the packet size. However, the CSMA/CA Markov chain defined later is still ergodic even without the assumption.

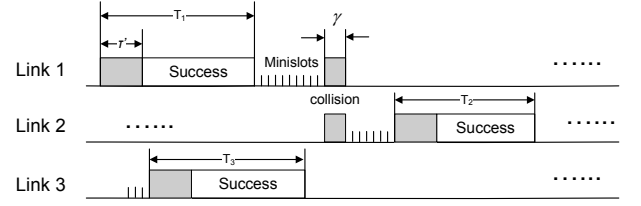


Fig. 3: Timeline in the basic model (In this figure, $\tau_i = T_i, i = 1, 2, 3$ are constants.)

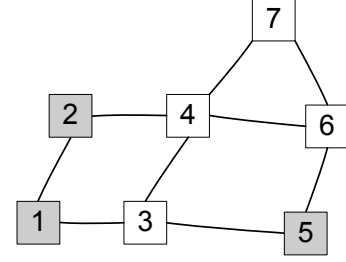


Fig. 4: An example conflict graph (each square represents a link). In this on-off state x , links 1, 2, 5 are active. So $S(x) = \{5\}$, $Z(x) = \{1, 2\}$, $h(x) = 1$.

(transmitting) in state x , and $x_k = 0$ otherwise. Thus, x is a vector indicating which links are active in a given slot. Let $G(x)$ be the subgraph of G after removing all vertices (each representing a link) with state 0 (i.e., any link j with $x_j = 0$) and their associated edges. In general, $G(x)$ is composed of a number of connected components (simply called “components”) $C_m(x), m = 1, 2, \dots, M(x)$ (where each component is a set of links, and $M(x)$ is the total number of components in $G(x)$). If a component $C_m(x)$ has only one active link (i.e., $|C_m(x)| = 1$), then this link is having a successful transmission; if $|C_m(x)| > 1$, then all the links in the component are experiencing a collision. Let the set of “successful” links in state x be $S(x) := \{k | k \in C_m(x) \text{ with } |C_m(x)| = 1\}$, and the set of links that are experiencing collisions be $Z(x)$. Also, define the “collision number” $h(x)$ as the number of components in $G(x)$ with size larger than 1. Fig. 4 shows an example. Note that the transmissions in a collision component $C_m(x)$ are “synchronized”, i.e., the links in $C_m(x)$ must have started transmitting in the same slot, and will end transmitting in the same slot after γ slots (the length of the probe packets).

C. Computation of the service rates

In order to compute the service rates of all the links under the above CSMA protocol when all the links are saturated, we first define the underlying discrete-time Markov chain which we call the *CSMA/CA Markov chain*.

The Markov chain evolves slot by slot.² The state of the Markov chain in a slot is

$$w := \{x, ((b_k, a_k), \forall k : x_k = 1)\} \quad (3)$$

where b_k is the total length of the current packet link k is transmitting, a_k is the remaining time (including the current

²For the ease of analysis, we make the modeling assumption that the links are synchronized at the slot level.

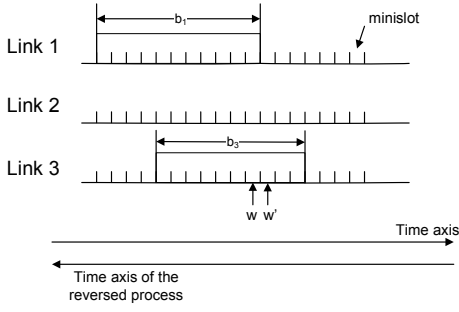


Fig. 5: Example of the CSMA/CA Markov chain

slot) before the transmission of link k ends.

For example, in Fig. 5, the state w and w' are

$$w = \{x = (1, 0, 1)^T, (b_1 = 11, a_1 = 1), (b_3 = 10, a_3 = 4)\} \quad (4)$$

and

$$w' = \{x = (0, 0, 1)^T, (b'_3 = 10, a'_3 = 3)\}. \quad (5)$$

Note that in any state w as defined in (3), we have

(I) $1 \leq a_k \leq b_k, \forall k : x_k = 1$.

(II) $P_k(b_k) > 0, \forall k \in S(x)$.

(III) If $k \in Z(x)$, then $b_k = \gamma$ and $a_k \in \{1, 2, \dots, \gamma\}$.

An important observation here is that the transmissions in a collision component $C_m(x)$ are “synchronized”, i.e., the links in $C_m(x)$ must have started transmitting at the same time, and will end transmitting at the same time, so all links in the component $C_m(x)$ have the same remaining time. (To see this, first note that in the case of a collision only the probe packets get transmitted, and their transmission times γ are identical for all links. Second, any two links i and j in this component with an edge between them must have started transmitting at the same time. Otherwise, if i starts earlier, j would not transmit since it already hears i ’s transmission; and vice versa. By induction, all links in the component must have started transmitting at the same time.) So, we can write $a_k = a^{(m)}$ for any $k \in C_m(x)$ where $|C_m(x)| > 1$, and $a^{(m)}$ denotes the remaining time of the component $C_m(x)$.

We say that a state w is *valid* iff it satisfies (I)–(III) above.

Since the transmission lengths are always bounded by b_{max} by assumption, we have $b_k \leq b_{max}$, and therefore the Markov chain has a finite number of states, and is ergodic. As detailed in Appendix A, a nice property of this Markov chain is that it is “almost” time-reversible. As a result, its stationary distribution has a simple product-form (Appendix A), from which the probability of any on-off state x can be computed:

Theorem 1: Under the stationary distribution, the probability of $x \in \{0, 1\}^K$ in a given slot is

$$\begin{aligned} p(x) &= \frac{1}{E} (\gamma^{h(x)} \prod_{k \in S(x)} T_k) \prod_{i: x_i=0} (1 - p_i) \prod_{j: x_j=1} p_j \\ &= \frac{1}{E} (\gamma^{h(x)} \prod_{k \in S(x)} T_k) \prod_{i=1}^K p_i^{x_i} q_i^{1-x_i} \end{aligned} \quad (6)$$

where $q_i := 1 - p_i$, T_i is the mean transmission length of link i (as defined in (2)), and E is a normalizing term such that

$$\sum_{x \in \{0, 1\}^K} p(x) = 1.^3$$

The proof is given in Appendix A.⁴

Remark: Note that in x , some links can be in a collision state, just as in IEEE 802.11. This is reflected in the $\gamma^{h(x)}$ term in (6). Expression (6) differs from the idealized-CSMA case in [16] and the stationary distribution in the data phase in the protocol proposed in [8], due to the difference of the three protocols.

Now we re-parametrize T_k by a variable r_k . Let $T_k := \tau' + T_0 \cdot \exp(r_k)$, where τ' , as we defined, is the overhead of a successful transmission (including RTS, CTS, ACK packets, DIFS, etc.), and $T_k^p := T_0 \cdot \exp(r_k)$ is the mean length of the payload. $T_0 > 0$ is a constant “reference payload length”. Let \mathbf{r} be the vector of r_k ’s. By Theorem 1, the stationary probability of x in a slot (with a given \mathbf{r}) is

$$p(x; \mathbf{r}) = \frac{1}{E(\mathbf{r})} g(x) \cdot \prod_{k \in S(x)} (\tau' + T_0 \cdot \exp(r_k)) \quad (7)$$

where $g(x) = \gamma^{h(x)} \prod_{i=1}^K p_i^{x_i} q_i^{1-x_i}$ is not related to \mathbf{r} , and the normalizing term is

$$E(\mathbf{r}) = \sum_{x' \in \{0, 1\}^K} [g(x') \cdot \prod_{k \in S(x')} (\tau' + T_0 \cdot \exp(r_k))]. \quad (8)$$

Then, the stationary probability that link k is transmitting a payload in a given slot is

$$s_k(\mathbf{r}) = \frac{T_0 \cdot \exp(r_k)}{\tau' + T_0 \cdot \exp(r_k)} \sum_{x: k \in S(x)} p(x; \mathbf{r}). \quad (9)$$

Recall that the capacity of each link is 1. Also, it’s easy to show that the CSMA/CA Markov chain is ergodic. As a result, if \mathbf{r} is fixed, the long-term average throughput of link k converges to the stationary probability $s_k(\mathbf{r})$. So we say that $s_k(\mathbf{r}) \in [0, 1]$ is the *service rate* of link k .

III. A DISTRIBUTED ALGORITHM TO APPROACH THROUGHPUT-OPTIMALITY

A. The scheduling problem

Assume that the conflict graph G has N different independent sets (“IS”, not confined to “maximal independent sets”), where each IS is a set of links that can transmit simultaneously without conflict. Denote an IS by $\sigma \in \{0, 1\}^K$, a 0-1 vector that indicates which links are transmitting in this IS. The k ’th element of σ , $\sigma_k = 1$ if link k is transmitting in this IS, and $\sigma_k = 0$ otherwise. Let \mathcal{X} be the set of ISs.

We now describe the *scheduling problem* which is the focus of the paper. Traffic arrives at link k with an arrival rate $\lambda_k \in (0, 1)$. For simplicity, assume the following i.i.d. Bernoulli arrivals (although this can be easily generalized): at the beginning of slot $M \cdot i$, ($i = 0, 1, 2, \dots$), a packet with a

³In this paper, several kinds of “states” are defined. With a little abuse of notation, we always use $p(\cdot)$ to denote the probability of the “state” under the stationary distribution of the CSMA/CA Markov chain. This does not cause confusion since the meaning of $p(\cdot)$ is clear from its argument.

⁴In [20], a similar model for CSMA/CA network is formulated with analogy to a loss network [21]. However, since [20] studied the case when the links are unsaturated, the explicit expression of the stationary distribution was difficult to obtain.

length of M slots arrives at link k with probability λ_k . (That is, the packet would take M slots to transmit.) Clearly, link k needs to be active with a probability at least λ_k to serve the arrivals. Denote the vector of arrival rates by $\lambda \in \mathcal{R}_+^K$.

Since we focus on the scheduling problem, all the packets traverse only one link (i.e., *single-hop*) before they leave the network. However, the results here can be extended to multi-hop networks and be combined with congestion control as in [16].

Definition 1: We say that λ is *feasible* iff it can be written as $\lambda = \sum_{\sigma \in \mathcal{X}} [\bar{p}_\sigma \cdot \sigma]$ where $\bar{p}_\sigma \geq 0$ and $\sum_{\sigma \in \mathcal{X}} \bar{p}_\sigma = 1$. That is, there is a schedule of the independent sets (including the non-maximal ones) that can serve the arrivals. Denote the set of feasible λ by $\bar{\mathcal{C}}$. We say that λ is *strictly feasible* iff $\lambda \in \mathcal{C}$, where \mathcal{C} is the interior of $\bar{\mathcal{C}}$.⁵

A scheduling algorithm is said to be “throughput-optimal” if it can “support” any $\lambda \in \mathcal{C}$. In this paper, this means that for any $\lambda \in \mathcal{C}$, the scheduling algorithm can provide to link k a service rate at least λ_k for all k .

B. CSMA scheduling with collisions

The following theorem states that any service rates equal to $\lambda \in \mathcal{C}$ can be achieved by properly choosing the mean payload lengths $T_k^p := T_0 \exp(r_k)$, $\forall k$.

Theorem 2: Assume that $\gamma, \tau' > 0$, and transmission probabilities $p_k \in (0, 1)$, $\forall k$ are fixed. Given any $\lambda \in \mathcal{C}$, there exists a unique $\mathbf{r}^* \in \mathcal{R}^K$ such that the service rate of link k is equal to the arrival rate for all k :

$$s_k(\mathbf{r}^*) = \lambda_k, \forall k. \quad (10)$$

Moreover, \mathbf{r}^* is the solution of the convex optimization problem

$$\max_{\mathbf{r}} L(\mathbf{r}; \lambda) \quad (11)$$

where

$$L(\mathbf{r}; \lambda) = \sum_k (\lambda_k r_k) - \log(E(\mathbf{r})), \quad (12)$$

with $E(\mathbf{r})$ defined in (8). This is because $\partial L(\mathbf{r}; \lambda) / \partial r_k = \lambda_k - s_k(\mathbf{r})$, $\forall k$.

The proof is in Appendix B.

Theorem 2 motivates us to design a gradient algorithm to solve problem (11). However, due to the randomness of the system, λ_k and $s_k(\mathbf{r})$ cannot be obtained directly and need to be estimated. We design the following distributed algorithm, where each link k dynamically adjusts its mean payload length T_k^p based on local information.

Algorithm 1: Transmission length control algorithm

The vector \mathbf{r} is updated every M slots. Specifically, it is updated at the beginning of slot $M \cdot i$, $i = 1, 2, \dots$. Denote by t_i the time when slot $M \cdot i$ begins. Also define $t_0 = 0$. Let “period i ” be the time between t_{i-1} and t_i , and $\mathbf{r}(i)$ be the value of \mathbf{r} at the end of period i , i.e., at time t_i . Initially, link k sets $r_k(0) \in [r_{\min}, r_{\max}]$ where r_{\min}, r_{\max} are two

parameters (to be further discussed). Then at time t_i , $i = 1, 2, \dots$, each link k updates

$$r_k(i) = r_k(i-1) + \alpha(i) [\lambda'_k(i) - s'_k(i) + \tilde{h}(r_k(i-1))] \quad (13)$$

where $\tilde{h}(\cdot)$ is a “penalty function” to be defined later, $\alpha(i) > 0$ is the step size in period i , $\lambda'_k(i)$, $s'_k(i)$ are the empirical average arrival rate and service rate in period i (i.e., the actual amount of arrived traffic and served traffic in period i divided by M).

The use of dummy bits: An important point here is that we let link k add dummy bits to the payload when its queue has less bits than what is specified by the algorithm (e.g., in (15) below). If the queue is empty, then dummy packets are transmitted with the specified size. So, each link is *saturated*. This ensures that the CSMA/CA Markov chain has the desired stationary distribution in (6). The transmitted dummy bits are also included in the computation of $s'_k(i)$. (Although the use of dummy bits consumes bandwidth, it simplifies our analysis, and does not prevent us from achieving the primary goal, i.e., approaching throughput-optimality. In Section V-B, we also simulate the case without dummy bits.)

Note that $\lambda'_k(i)$, $s'_k(i)$ are random variables which are generally not equal to λ_k and $s_k(\mathbf{r}(i-1))$. Assume that the maximal instantaneous arrival rate is $\bar{\lambda}$, so $\lambda'_k(i) \leq \bar{\lambda}$, $\forall k, i$.

Also, in (13), the penalty function $\tilde{h}(\cdot)$ is defined as

$$\tilde{h}(y) = \begin{cases} r_{\min} - y & \text{if } y < r_{\min} \\ 0 & \text{if } y \in [r_{\min}, r_{\max}] \\ r_{\max} - y & \text{if } y > r_{\max}. \end{cases} \quad (14)$$

As shown in the Appendix, this function keeps $\mathbf{r}(i)$ in a bounded region. (This is a “softer” approach than directly projecting $r_k(i)$ to the set $[r_{\min}, r_{\max}]$. The purpose is only to simplify the proof of Theorem 3 later.)

In period $i + 1$, given $\mathbf{r}(i)$, we need to choose $\tau_k^p(i)$, the payload lengths of each link k , so that $E(\tau_k^p(i)) = T_k^p(i) = T_0 \exp(r_k(i))$. If $T_k^p(i)$ is an integer, then we let $\tau_k^p(i) = T_k^p(i)$; otherwise, we randomize $\tau_k^p(i)$ as follows:

$$\tau_k^p(i) = \begin{cases} \lceil T_k^p(i) \rceil & \text{with probability } T_k^p(i) - \lfloor T_k^p(i) \rfloor \\ \lfloor T_k^p(i) \rfloor & \text{with probability } \lceil T_k^p(i) \rceil - T_k^p(i). \end{cases} \quad (15)$$

Here, for simplicity, we have assumed that the arrived packets can be fragmented and reassembled to obtain the desired lengths $\lceil T_k^p(i) \rceil$ or $\lfloor T_k^p(i) \rfloor$. However, *one can avoid the fragmentation* by randomizing the number of transmitted packets (each with a length of M slots) in a similar way. When there are not enough bits in the queue, “dummy bits” are generated (as mentioned before) to satisfy $E(\tau_k^p(i)) = T_0 \exp(r_k(i))$ and make the links always saturated.

Intuitively speaking, Algorithm 1 says that when $r_k \in [r_{\min}, r_{\max}]$, if the empirical arrival rate of link k is larger than the service rate, then link k should transmit more aggressively by using a larger mean transmission length, and vice versa.

Algorithm 1 is parametrized by r_{\min}, r_{\max} which are fixed during the execution of the algorithm. Note that the choice of r_{\max} affects the maximal possible payload length. Also, as

⁵That is, $\mathcal{C} := \{\lambda' \in \bar{\mathcal{C}} | \mathcal{B}(\lambda', d) \subseteq \bar{\mathcal{C}} \text{ for some } d > 0\}$, where $\mathcal{B}(\lambda', d) = \{\tilde{\lambda} | \|\tilde{\lambda} - \lambda'\|_2 \leq d\}$ is a ball centered at λ' with radius d .

discussed below, the choices of r_{max} and r_{min} also determine the “capacity region” of Algorithm 1.

We define the region of arrival rates

$$\mathcal{C}(r_{min}, r_{max}) := \{\lambda \in \mathcal{C} | \mathbf{r}^*(\lambda) \in (r_{min}, r_{max})^K\} \quad (16)$$

where $\mathbf{r}^*(\lambda)$ denotes the unique solution of $\max_{\mathbf{r}} L(\mathbf{r}; \lambda)$ (such that $s_k(\mathbf{r}^*) = \lambda_k, \forall k$, by Theorem 2). Later we show that the algorithm can “support” any $\lambda \in \mathcal{C}(r_{min}, r_{max})$ in some sense under certain conditions on the step sizes. We will also give a characterization of the region $\mathcal{C}(r_{min}, r_{max})$ later in section IV.

Clearly, $\mathcal{C}(r_{min}, r_{max}) \rightarrow \mathcal{C}$ as $r_{min} \rightarrow -\infty$ and $r_{max} \rightarrow \infty$, where \mathcal{C} is the set of all strictly feasible λ (by Theorem 2). Therefore, although given (r_{min}, r_{max}) the region $\mathcal{C}(r_{min}, r_{max})$ is smaller than \mathcal{C} , one can choose (r_{min}, r_{max}) to arbitrarily approach the maximal capacity region \mathcal{C} . Also, there is a tradeoff between the capacity region and the maximal packet length.

Theorem 3: Assume that the vector of arrival rates $\lambda \in \mathcal{C}(r_{min}, r_{max})$. With Algorithm 1,

(i) If $\alpha(i) > 0$ is non-increasing and satisfies $\sum_i \alpha(i) = \infty$, $\sum_i \alpha(i)^2 < \infty$ and $\alpha(1) \leq 1$ (for example, $\alpha(i) = 1/i$), then $\mathbf{r}(i) \rightarrow \mathbf{r}^*$ as $i \rightarrow \infty$ with probability 1, where \mathbf{r}^* satisfies $s_k(\mathbf{r}^*) = \lambda_k, \forall k$.

(ii) The case with constant step size (i.e., $\alpha(i) = \alpha, \forall i$): For any $\delta > 0$, there exists a small enough $\alpha > 0$ such that $\liminf_{J \rightarrow \infty} \sum_{i=1}^J s'_k(i)/J \geq \lambda_k - \delta, \forall k$ with probability 1. In other words, one can achieve average service rates arbitrarily close to the arrival rates by choosing α small enough.

Remark: In [8] which proposed an alternative algorithm to deal with collisions, the authors made an idealized time-scale-separation assumption that the CSMA/CA Markov chain reaches its stationary distribution for any given CSMA parameters. We believe that the results in Theorem 3 can be extended to their algorithm.

The complete proof of Theorem 3 is Appendix C, but the result can be intuitively understood as follows. If the step size is small (in (i), $\alpha(i)$ becomes small when i is large), r_k is “quasi-static” such that roughly, the service rate is averaged (over multiple periods) to $s_k(\mathbf{r})$, and the arrival rate is averaged to λ_k . Thus the algorithm solves the optimization problem (11) by a stochastic approximation [23] argument, such that $\mathbf{r}(i)$ converges to \mathbf{r}^* in part (i), and $r(i)$ is near \mathbf{r}^* with high probability in part (ii).

Corollary 1: Consider a variant of Algorithm 1 below where the update equation of each link k is

$$r_k(i) = r_k(i-1) + \alpha(i)[\lambda'_k(i) + \Delta - s'_k(i) + \tilde{h}(r_k(i-1))] \quad (17)$$

with a small constant $\Delta > 0$. That is, the algorithm “pretends” to serve the arrival rate $\lambda + \Delta \cdot \mathbf{1}$ which is slightly larger than the actual λ . Assume that

$$\begin{aligned} \lambda &\in \mathcal{C}'(r_{min}, r_{max}, \Delta) \\ &:= \{\lambda | \lambda + \Delta \cdot \mathbf{1} \in \mathcal{C}(r_{min}, r_{max})\}. \end{aligned}$$

For algorithm (17), one has the following results:

(i) if $\alpha(i) > 0$ is non-increasing and satisfies $\sum_i \alpha(i) = \infty$, $\sum_i \alpha(i)^2 < \infty$ and $\alpha(1) \leq 1$ (for example, $\alpha(i) = 1/i$), then

$\mathbf{r}(i) \rightarrow \mathbf{r}^*$ as $i \rightarrow \infty$ with probability 1, where \mathbf{r}^* satisfies $s_k(\mathbf{r}^*) = \lambda_k + \Delta > \lambda_k, \forall k$;

(ii) if $\alpha(i) = \alpha$ (i.e., constant step size) where α is small enough, then all queues are positive recurrent (and therefore stable).

Algorithm (17) is parametrized by r_{min}, r_{max} and Δ . Clearly, as $r_{min} \rightarrow -\infty$, $r_{max} \rightarrow \infty$ and $\Delta \rightarrow 0$, $\mathcal{C}'(r_{min}, r_{max}, \Delta) \rightarrow \mathcal{C}$, the maximal capacity region.

The proof is similar to that of Theorem 3 and is given in [30]. A sketch is as follows: Part (i) is similar to (i) in Theorem 3. The extra fact that $s_k(\mathbf{r}^*) > \lambda_k, \forall k$ reduces the queue size compared to Algorithm 1 (since when the queue size is large enough, it tends to decrease). Part (ii) holds because if we choose $\delta = \Delta/2$, then by Theorem 3, $\liminf_{J \rightarrow \infty} \sum_{i=1}^J s'_k(i)/J \geq \lambda_k + \Delta - \delta > \lambda_k, \forall k$ almost surely if α is small enough. Then the result follows by showing that the queue sizes have negative drift.

IV. RELATIONSHIP BETWEEN THE CSMA PARAMETERS AND THE CAPACITY REGION

In the previous section, we mentioned that the region $\mathcal{C}(r_{min}, r_{max})$ (and $\mathcal{C}'(r_{min}, r_{max}, \Delta)$) becomes larger as we decrease r_{min} and/or increase r_{max} . Therefore, fixing r_{min} , a larger r_{max} leads to a larger capacity region, but allows for larger transmission lengths. In practice, however, the transmission lengths should not be too long, since longer transmission lengths lead to larger access delay (where the access delay refers to the time between the beginnings of two consecutive successful transmissions of a link) and larger variations of the delay, and consequently, poorer *short-term fairness*. It is especially the case when a link has a number of conflicting links which do not interfere with each other. Then the link has to wait for all the conflicting links to become inactive before attempting its transmission. This issue has been studied in [26][22][27] in the contexts of 1-D and 2-D lattice topologies, and star topologies, where it is shown that the short-term fairness worsens when the *access intensities* (i.e., the ratios between the average transmission times and mean backoff times) increase⁶. Although references [26][22][27] focus on the collision-free idealized-CSMA, we observe the same phenomenon in the simulations of our model (see Appendix E for some simulation results in the 1-D and 2-D lattice topologies).

Therefore, there is a *tradeoff* between the long-term efficiency (i.e., the capacity region) and short-term fairness. To quantify the tradeoff we need to understand two relationships. The first is the relationship between the maximal required transmission lengths and the capacity region. And the second is between the maximal transmission lengths and the short-term fairness.

We first discuss the second relationship. For simplicity, assume the arrival rate vector is λ , and that Algorithm 1 has converged to the suitable mean payload lengths $T_k^p := T_0 \exp(r_k^*(\lambda)), \forall k$ (Recall that $\mathbf{r}^*(\lambda)$ is the vector such that

⁶Reference [26] also showed that when the access intensities are high, there exists long-term unfairness in the 2-D lattice topology under different boundary conditions.

$s_k(\mathbf{r}^*(\lambda)) = \lambda_k, \forall k$. Assume that we fix the mean payload lengths at T_k^p 's, and denote the (random) access delay of link k by D_k . We use two quantities to measure the short-term fairness of link k : the mean and standard deviation of D_k (i.e., $E(D_k)$ and $\sqrt{\text{var}(D_k)}$). Similar to [22], one has

$$E(D_k) = \frac{T_k^p}{s_k(\mathbf{r}^*(\lambda))} = \frac{T_k^p}{\lambda_k}, \forall k.$$

Therefore if we can find an upper bound of $T_k^p = T_0 \exp(r_k^*(\lambda))$ (to be further discussed in this section), then an upper bound of $E(D_k)$ can be obtained. (In fact, in Algorithm 1, the long-term average of D_k is also T_k^p/λ_k , since the initial convergence phase is not significant in the long term.) On the other hand, obtaining an expression of $\sqrt{\text{var}(D_k)}$ for general topologies is difficult and deserves future research. (In Section V-A we will present some numerical results.) Therefore, how to choose r_{max} to ensure that $\sqrt{\text{var}(D_k)}$ is lower than some threshold remains an open problem.

Next we consider the first relationship. We present several generic bounds to characterize how the regions $\mathcal{C}(r_{min}, r_{max})$ and $\mathcal{C}'(r_{min}, r_{max}, \Delta)$ depend on r_{max} and r_{min} . Given a $\lambda \in \mathcal{C}$, by the definition of $\mathcal{C}(r_{min}, r_{max})$ in (16), if one chooses $r_{min} < \min_k r_k^*(\lambda)$ and $r_{max} > \max_k r_k^*(\lambda)$, then $\lambda \in \mathcal{C}(r_{min}, r_{max})$, so that Algorithm 1 can be used to support λ . ($r_k^*(\lambda)$ is the k -th element of $\mathbf{r}^*(\lambda)$.) A similar statement can be made for $\mathcal{C}'(r_{min}, r_{max}, \Delta)$.

Consider a vector $\bar{\lambda} \succ \mathbf{0}$ which is at the boundary of $\bar{\mathcal{C}}$ (i.e., $\bar{\lambda} \in \bar{\mathcal{C}}$ but $\rho \bar{\lambda} \notin \bar{\mathcal{C}}, \forall \rho > 1$). Clearly, for $\rho \in (0, 1)$, $\rho \bar{\lambda} \in \bar{\mathcal{C}}$. Denote $\rho = 1 - \epsilon$. We are interested to bound $\mathbf{r}^*((1-\epsilon)\bar{\lambda})$. For the idealized CSMA model without collisions used in [16], an earlier bound obtained in [31] (Lemma 8-(3)) suggests that $\max_k r_k^*((1-\epsilon)\bar{\lambda}) \leq O(1/\epsilon)$ (where \mathbf{r}^* there controls the backoff times). In this section, we show a stronger result that, in our model with collisions, $\max_k r_k^*((1-\epsilon)\bar{\lambda}) \leq O(\log(1/\epsilon))$, so that the required r_{max} to support arrival rates $(1-\epsilon)\bar{\lambda}$ is not more than $O(\log(1/\epsilon))$. (Also, one can similarly show that the same order $O(\log(1/\epsilon))$ applies to the idealized CSMA model as well.)

Theorem 4: We have

$$\bar{\lambda}^T \mathbf{r}^*((1-\epsilon)\bar{\lambda}) \leq b \cdot \left[\log\left(\frac{1}{\epsilon}\right) + \log\left(\frac{N'}{b}\right) + 2G + 1 \right] \quad (18)$$

for some constants N' , b and G (defined during the proof), if $\epsilon \leq 1/b$. When $\epsilon \in (1/b, 1)$,

$$\bar{\lambda}^T \mathbf{r}^*((1-\epsilon)\bar{\lambda}) \leq [\log(N') + 2G]/\epsilon. \quad (19)$$

So roughly speaking, as $\epsilon \rightarrow 0$, the value of $\mathbf{r}^*(\lambda)$ is not more than $O(\log(1/\epsilon))$ by (18).

The proof is in Appendix D.

The following is a lower bound of $\mathbf{r}^*(\lambda)$.

Proposition 1: Given any $\lambda \in \mathcal{C}$, we have

$$r_k^*(\lambda) \geq \log\left(\frac{\tau'}{T_0} \frac{\lambda_k}{1 - \lambda_k}\right), \forall k. \quad (20)$$

Therefore

$$\min_k r_k^*(\lambda) \geq \log\left(\frac{\tau'}{T_0} \frac{\min_k \lambda_k}{1 - \min_k \lambda_k}\right). \quad (21)$$

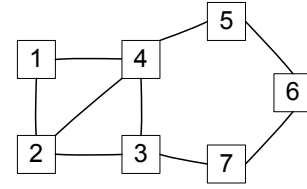


Fig. 6: The conflict graph in simulations

Proof: Suppose that $r_k^*(\lambda) < \log(\frac{\tau'}{T_0} \frac{\lambda_k}{1 - \lambda_k})$, then the mean payload length is $T_0 \exp(r_k^*(\lambda)) < \tau' \cdot \lambda_k / (1 - \lambda_k)$. Note that the overhead of each successful transmission is τ' . So, even if link k is successfully transmitting all the time, its service rate would be strictly less than $\tau' \cdot \frac{\lambda_k}{1 - \lambda_k} / (\tau' + \tau' \cdot \frac{\lambda_k}{1 - \lambda_k}) = \lambda_k$, leading to a contradiction. ■

Then, we have the following result.

Corollary 2: $\max_k r_k^*((1-\epsilon)\bar{\lambda}) \leq O(\log(1/\epsilon))$ as $\epsilon \rightarrow 0$.

Proof: For convenience, denote $\lambda = (1-\epsilon)\bar{\lambda}$. Since we are interested in the asymptotic behavior as $\epsilon \rightarrow 0$, assume that $\epsilon \leq 0.5$. Denote $\bar{\lambda}_{min} := \min_k \bar{\lambda}_k > 0$. Then, $\lambda_k \geq 0.5\bar{\lambda}_{min}, \forall k$.

By (20), we know that $r_k^*(\lambda) \geq \log(\frac{\tau'}{T_0}) + \log(\frac{\lambda_k}{1 - \lambda_k}) \geq \log(\frac{\tau'}{T_0}) + \log(\frac{0.5\bar{\lambda}_{min}}{1 - 0.5\bar{\lambda}_{min}}) := \underline{r}, \forall k$. Then, combined with (18), if $\epsilon \leq 1/b$, we have for any k ,

$$\begin{aligned} r_k^*(\lambda) &\leq \{b \cdot [\log(\frac{N'}{b \cdot \epsilon}) + 2G + 1] - \sum_{k' \neq k} \bar{\lambda}_{k'} \cdot r_{k'}^*(\lambda)\} / \bar{\lambda}_k \\ &\leq \{b \cdot [\log(\frac{N'}{b \cdot \epsilon}) + 2G + 1] - \sum_{k' \neq k} \bar{\lambda}_{k'} \cdot \underline{r}\} / \bar{\lambda}_k \\ &= O(\log(1/\epsilon)) \end{aligned}$$

which completes the proof. ■

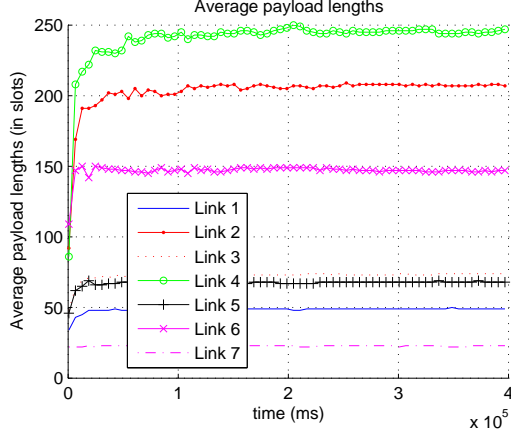
V. NUMERICAL RESULTS

Consider the conflict graph in Fig. 6. Let the vector of arrival rates be $\lambda = \rho \cdot \bar{\lambda}$, where $\rho \in (0, 1)$ is the “load”, and $\bar{\lambda}$ is a convex combination of several maximal IS: $\bar{\lambda} = 0.2 * [1, 0, 1, 0, 1, 0, 0] + 0.2 * [0, 1, 0, 0, 1, 0, 1] + 0.2 * [0, 0, 0, 1, 0, 1, 0] + 0.2 * [0, 1, 0, 0, 0, 1, 0] + 0.2 * [1, 0, 1, 0, 0, 1, 0] = [0.4, 0.4, 0.4, 0.2, 0.4, 0.6, 0.2]$. Since $\rho \in (0, 1)$, λ is strictly feasible. Fix the transmission probabilities as $p_k = 1/16, \forall k$. The “reference payload length” $T_0 = 15$.

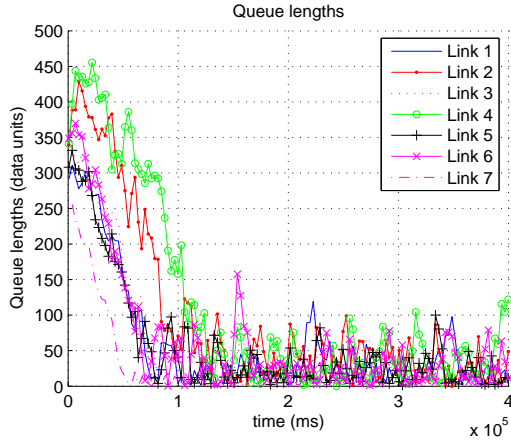
A. Transmission length control algorithms

We evaluate algorithm (17) (a variant of Algorithm 1) in our C++ simulator. The update in (17) is performed every $M = 500$ slots. Let the step size $\alpha(i) = 0.23/(2 + i/100)$, the upper bound $r_{max} = 3.5$, the lower bound $r_{min} = 0$, and the “gap” $\Delta = 0.005$. The initial value of each r_k is 0.

Let the “load” of arrival rates be $\rho = 0.8$ (i.e., $\lambda = 0.8 \cdot \bar{\lambda}$). The collision length (e.g., RTS length) is $\gamma = 5$, and the overhead of successful transmission is $\tau' = 10$. To show the negative drift of the queue lengths, assume that initially all queue lengths are 300 data units (where each data unit takes 100 slots to transmit). As expected, Fig. 7 (a) shows the convergence of the mean payload lengths, and Fig. 7 (b) shows that all queues are stable.



(a) Convergence of the mean payload lengths



(b) Stability of the queues

Fig. 7: Simulation of Algorithm (17) (with the conflict graph in Fig. 6)

ρ	0.65	0.7	0.75	0.8	0.85
Mean of D_3 (in slots)	125.9	146.7	178.3	226.3	310.3
Standard deviation of D_3	132.7	161.9	211.5	293.7	456.1

TABLE I: Short-term fairness of link 3

	R_1	R_2	R_3	R_4	R_5	R_6
$\theta = 0.15$ (Simulation)	0.279	0.386	0.547	0.548	0.387	0.279
$\theta = 0.15$ ([28])	0.272	0.347	0.442	0.442	0.347	0.273
$\theta = 0.2$ (Simulation)	0.526	0.837	1.372	1.371	0.840	0.526
$\theta = 0.2$ ([28])	0.5	0.75	1.125	1.125	0.75	0.5
$\theta = 0.25$ (Simulation)	1.075	2.229	4.735	4.733	2.240	1.072
$\theta = 0.25$ ([28])	1	2	4	4	2	1
$\theta = 0.3$ (Simulation)	3.210	12.94	52.76	52.32	12.91	3.209
$\theta = 0.3$ ([28])	3	12	48	48	12	3

TABLE II: Comparison of access intensities

To study the tradeoff between the load ρ and short-term fairness, we run Algorithm (17) for $\rho \in \{0.65, 0.7, 0.75, 0.8, 0.85\}$. In each case, we collect the data of the access delay and compute its mean and standard deviation when \mathbf{r} has almost converged. Table I shows the results for link 3 (and other links have a similar trend). Note that when ρ increases, both the standard deviation and the mean increase, and their ratio increases too, indicating poorer short-term fairness.

In [28], van de Ven et al. considered the line topology (i.e., 1-D lattice topology) and obtained the explicit expression of the access intensity of each link required to support a uniform throughput θ for all the links, under the idealized-CSMA model without collisions [14], [25], [26], [27]. For comparison, we simulate Algorithm 1 in a line topology with 6 links, where each link conflicts with the first 2 links on both sides. After Algorithm 1 converges, we compute the access intensity of link k as $R_k := T_k^p / (1/p_k - 1)$ (since the mean backoff time of link k is $1/p_k - 1$), and compare it to the result of Theorem 2 in [28] (although the “access intensities” under the two models are not completely equivalent due to our inclusion of collisions.) Let $p_k = 1/16, \forall k$, $\gamma = 1$ and $\tau' = 1$. We simulate four sets of arrival rates, $\lambda = \theta \cdot 1$ where $\theta = 0.15, 0.2, 0.25$ and 0.3 , and give the results in Table II.

The results show a close match, with relatively larger differences in R_3 and R_4 . The reason is that link 3 and 4 are in the middle of the network and suffer from more collisions. After each collision link 3 (or 4) needs to restart the backoff, which increases its effective backoff time and therefore requires a larger payload length to compensate. Also, all R_k 's are higher in the simulation due to collisions and the overhead τ' .

B. Effect of dummy bits

We have used dummy bits to facilitate our analysis and design of the algorithms. However, transmitting dummy bits when a queue is empty consumes extra bandwidth. In this subsection, we simulate our algorithms without dummy bits.

In both Algorithm 1 and Algorithm (17), we make the following heuristic modification. For each link k , if $\tau_k^p(i)$ as computed in (15) is larger than the current (positive) queue length, then transmit a packet that includes all the bits of the queue as the payload. That is, no dummy bits are added. If the queue is empty then the link keeps silent. In the computation of $s_k'(i)$, however, the payload of the packet is counted as $\tau_k^p(i)$.⁷ Not surprisingly, the modified algorithms are difficult to analyze, and we therefore do not claim their convergence. (However, they still seem to converge in the simulations.)

Fig. 8 shows the evolution of the average payload lengths under Algorithm (17) without dummy bits, when $\rho = 0.8$. Indeed, the required payload lengths are significantly reduced compared to Fig. 7 (a) due to the saved bandwidth.

Under Algorithm 1, however, we find that the required payload lengths are very close with or without dummy bits.

⁷The reason for this design is that, if we only count the actual bits transmitted, then Algorithm (17) could not converge. Indeed, if Algorithm (17) converges, then the average service rates is strictly larger than the arrival rates, which is impossible if we only count the actually transmitted bits.

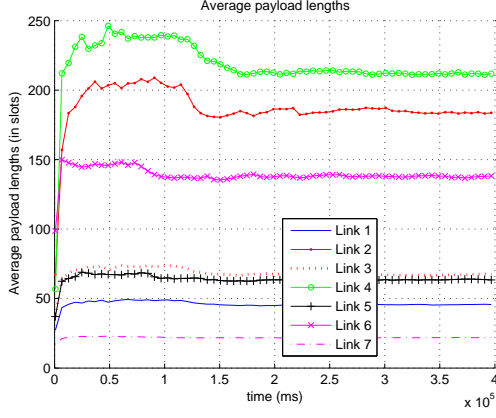


Fig. 8: Algorithm (17) without dummy bits

The reason is that, since Algorithm 1 only tries to make the average service rates equals the arrival rates, the queues do not have a negative drift towards zero. As a result, the queues are not close to zero most of the time, so dummy bits are rarely generated even if they are allowed.

C. Effect of hidden nodes

So far we have assumed that there is no hidden node (HN) in the network. In this subsection we discuss the effect of HNs.

Consider a simple network with 2 links that are hidden from each other. That is, they cannot hear the transmissions of each other but a collision occurs if their transmissions overlap. Unlike the case without HNs, a link can start transmitting *in the middle of* the other link's transmission and cause a collision.

First, to explore how much service rates can be achieved in this scenario, we let the two links use the same, fixed payload length τ^p . Let $\gamma = 5$, $\tau' = 10$, and $p_k = 1/64, \forall k$. The two links receive the same service rate by symmetry. Fig. 9 shows the service rate of one link under different values of τ^p . Note that the maximal service rate per link is about 0.12, much less than 0.5 in the case without HNs. Also, when τ^p is large enough, further increasing τ^p decreases the service rates, because larger packets are more easily collided by the HN.

Then we simulate Algorithm 1 with arrival rates $\lambda_1 = \lambda_2 = 0.1 < 0.12$. We set $T_0 = 15$, $r_{min} = 0$, $r_{max} = 2.59$ so that the maximal payload is $T_0 \exp(r_{max}) = 200$ (slots), and $\alpha(i) = 0.14/(2 + i/100)$. Unlike the case without HNs, the results depend on the initial condition as shown in Fig. 10. For example, if the initial payload lengths of both links are 40 slots (which we call “initial condition 1”), then the mean payload lengths converge to the correct value (about 17.5). However, if the initial payloads are 80 slots (“initial condition 2”), then the mean payload lengths keep increasing (until reaching the maximal value) and cannot support the arrival rates. This can be explained by Fig. 9. Initial payload lengths of 40 slots achieve a per-link service rate higher than the arrival rate. By Algorithm 1, the payload lengths are decreased and eventually converge to the correct values. However, if initially the payload lengths are 80 slots, a per-link service rate lower than 0.1 is achieved. By Algorithm 1, both links increase their payload

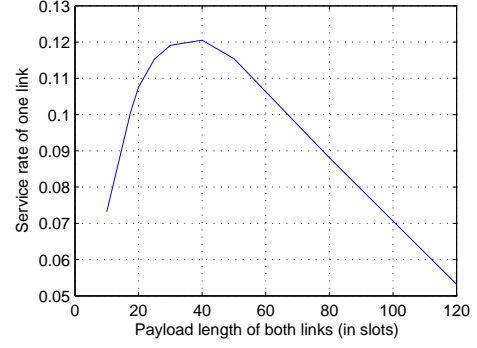


Fig. 9: Service rates in a 2-link network with hidden nodes

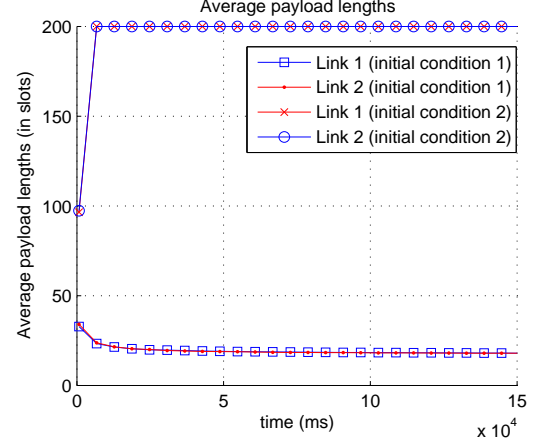


Fig. 10: Algorithm 1 with hidden nodes

lengths. This, however, further decreases their service rates, and the cycle goes on. The root cause for this behavior is as follows. Algorithm 1 has implicitly used the fact that, without HNs, a link's service rate increases with its payload length. However, it may not be the case when HNs exist.

To sum up, in the presence of HNs, both the achievable capacity region of CSMA and the property of our algorithms have changed. To address the HN problem, there are at least two directions to explore. The first is to understand the achievable capacity with HNs, and design algorithms to achieve the capacity. The second is to design protocols to remove or reduce HNs, so that our existing algorithms can be applied. There have been many proposals aiming to remove or reduce the HNs (see [19] and the references therein).

VI. CONCLUSION

In this paper, we have studied CSMA-based scheduling algorithms with collisions. We first provided a model and gave a throughput formula which takes into account the cost of collisions and overhead. The formula has a simple product form. Next, we designed distributed algorithms where each link adaptively updates its mean transmission length to approach the throughput-optimality, and provided sufficient conditions to ensure the convergence and stability of the algorithms. We also characterized the relationship between the algorithm parameters and the achievable capacity region.

Finally, simulations results were presented to illustrate and verify the main results.

In the algorithm, the transmission probabilities of the links are chosen to be fixed at a reasonable level, since we have shown that adjusting the transmission lengths alone is sufficient to approach throughput-optimality (the main goal of this paper). However, the choices of the transmission probabilities p_k 's has an effect on the probability of collisions among the probe packets. In the future, we would like to further study whether the adjustment of transmission probabilities can be combined with the algorithm. Also, we are interested to further explore the short-term fairness and the case with hidden nodes.

APPENDIX

A. Proof of Theorem 1

First, the stationary distribution of the CSMA/CA Markov chain is expressed in the following lemma.

Lemma 1: In the stationary distribution, the probability of a valid state w as defined by (3) is

$$p(w) = \frac{1}{E} \prod_{i:x_i=0} q_i \prod_{j:x_j=1} [p_j \cdot f(b_j, j, x)] \quad (22)$$

where

$$f(b_j, j, x) = \begin{cases} 1 & \text{if } j \in Z(x) \\ P_j(b_j) & \text{if } j \in S(x) \end{cases}, \quad (23)$$

where $P_j(b_j)$ is the p.m.f. of link j 's transmission length, as defined in (1). Also, E is a normalizing term such that $\sum_w p(w) = 1$, i.e., all probabilities sum up to 1. Note that $p(w)$ does not depend on the remaining time a_k 's.

Proof: For a given state $w = \{x, ((b_k, a_k), \forall k : x_k = 1)\}$, define the set of active links whose remaining time is larger than 1 as

$$A_1(w) = \{k | x_k = 1, a_k > 1\}.$$

Links in $A_1(w)$ will continue their transmissions (either with success or a collision) in the next slot.

Define the set of inactive links "blocked" by links in $A_1(w)$ as

$$\partial A_1(w) = \{j | x_j = 0; e(j, k) = 1 \text{ for some } k \in A_1(w)\}$$

where $e(j, k) = 1$ means that there is an edge between j and k in the conflict graph. Links in $\partial A_1(w)$ will remain inactive in the next slot.

Write $\bar{A}_1(w) := A_1(w) \cup \partial A_1(w)$. Define the set of all other links as

$$A_2(w) = \mathcal{N} \setminus \bar{A}_1(w). \quad (24)$$

These links can change their on-off states x_k 's in the next slot. On the other hand, links in $\bar{A}_1(w)$ will have the same on-off states x_k 's in the next slot.

To illustrate these notations, consider the example in Fig. 5. By (4), we have $A_1(w) = \{3\}$, $\partial A_1(w) = \{2\}$, $\bar{A}_1(w) = \{2, 3\}$ and $A_2(w) = \{1\}$.

State w can transit in the next slot to another valid state $w' = \{x', ((b'_k, a'_k), \forall k : x'_k = 1)\}$, i.e., $Q(w, w') > 0$, if and only if w' satisfies that (i) $x'_k = x_k, \forall k \in A_1(w)$; (ii) $b'_k = b_k, a'_k = a_k - 1, \forall k \in \bar{A}_1(w)$ such that $x_k = 1$; (iii)

$a'_k = b'_k, \forall k \in A_2(w)$ such that $x'_k = 1$, and $b'_k = \gamma, \forall k \in A_2(w) \cap Z(x')$. (If $A_2(w)$ is an empty set, then condition (iii) is trivially true.) The transition probability is

$$Q(w, w') = \prod_{i \in A_2(w)} [p_i \cdot f(b'_i, i, x')]^{x'_i} q_i^{1-x'_i}. \quad (25)$$

Define

$$\tilde{Q}(w', w) := \prod_{i \in A_2(w)} [p_i \cdot f(b_i, i, x)]^{x_i} q_i^{1-x_i}. \quad (26)$$

(If $A_2(w)$ is an empty set, then $Q(w, w') = 1$ and $\tilde{Q}(w', w) := 1$.) If w and w' does not satisfy all the conditions (i), (ii), (iii), then $Q(w, w') = 0$, and also define $\tilde{Q}(w', w) = 0$.

Then, if $Q(w, w') > 0$ (and $\tilde{Q}(w', w) > 0$), $p(w)/\tilde{Q}(w', w) = \frac{1}{E} \prod_{i \notin A_2(w)} [p_i \cdot f(b_i, i, x)]^{x_i} q_i^{1-x_i}$. And $p(w')/Q(w, w') = \frac{1}{E} \prod_{i \notin A_2(w)} [p_i \cdot f(b'_i, i, x')]^{x'_i} q_i^{1-x'_i}$. But for any $i \notin A_2(w)$, i.e., $i \in A_1(w)$, we have $x'_i = x_i, b'_i = b_i$ by condition (i), (ii) above. Therefore, the two expressions are equal. Thus

$$p(w)Q(w, w') = p(w')\tilde{Q}(w', w). \quad (27)$$

If two states w, w' satisfy $Q(w, w') = 0$, then by definition $\tilde{Q}(w', w) = 0$, making (27) trivially true. Therefore, (27) holds for any w and w' .

We will show later that $\tilde{Q}(w', w)$ is the transition probability of the "time-reversed process" of the above Markov chain (notice the similarity between $Q(w, w')$ and $\tilde{Q}(w', w)$), and naturally satisfies $\sum_w \tilde{Q}(w', w) = 1$. Assuming that the claim is true, then by (27), we have

$$\sum_w p(w) \cdot Q(w, w') = p(w') \sum_w \tilde{Q}(w', w) = p(w').$$

Therefore, $p(w)$ is the stationary (or "invariant") distribution, which completes the proof of Lemma 1.

It remains to be shown that the above claim is true, in particular, that $\sum_w \tilde{Q}(w', w) = 1$. Denote the original process as $\{w(t), t \in \mathcal{Z}\}$ (this is the Markov process that describes our CSMA protocol, with transition probabilities $Q(\cdot, \cdot)$ in (25)). Adding a time index in (3), we have

$$w(t) := \{x(t), ((b_k(t), a_k(t)), \forall k : x_k(t) = 1)\}. \quad (28)$$

Now define the time-reversed process $\tilde{w}(t) := w(-t), \forall t \in \mathcal{Z}$. First, note that in the process $\{w(t)\}$, the remaining time $a_k(t)$, if defined, decreases with t ; in the reversed process $\{\tilde{w}(t)\}$, however, $a_k(-t)$ increases with t . Therefore, $\{w(t)\}$ and $\{\tilde{w}(t)\}$ are, clearly, statistically distinguishable. So $\{w(t)\}$ is not time-reversible.

However, if we re-label the "remaining time" in the reversed order, then the process $\{\tilde{w}(t)\}$ "looks a lot" like the process $\{w(t)\}$. (This is why we say the CSMA/CA Markov chain is almost time-reversible.) More formally, with the understanding that $w = \{x, ((b_k, a_k), \forall k : x_k = 1)\}$, define a function $g(\cdot)$ as

$$g(w) = \{x, ((b_k, b_k - a_k + 1), \forall k : x_k = 1)\}. \quad (29)$$

Then define the process

$$\hat{w}(t) := g(w(-t)).$$

Note that in the process $\{\hat{w}(t)\}$, the “remaining time” decreases with t , similar to $\{w(t)\}$.

Next we show the following two facts.

Fact 1: For any two states w and w' with $Q(w, w') > 0$ (i.e., if the CSMA Markov chain can transit from state w to state w'), we have $A_1(w) = A_1(g(w'))$, $\bar{A}_1(w) = \bar{A}_1(g(w'))$ and $A_2(w) = A_2(g(w'))$.

Fact 2: $Q(w, w') > 0 \Leftrightarrow Q(g(w'), g(w)) > 0$.

These facts can be illustrated by the example in Fig. 5. First consider Fact 1. Note that $A_1(w) = \{3\}$, by definition, is the set of links that are in the middle of a transmission in state w and will continue the transmission in the next state w' . Then, in the reversed process, such links are also in the middle of a transmission in state $g(w')$ and will continue the transmission in the next state $g(w)$. So $A_1(w) = A_1(g(w'))$. Similarly, $\partial A_1(w) = \{2\}$, the set of links that are blocked by $A_1(w)$ in w are also blocked by $A_1(g(w'))$ in the reversed process. Therefore $\partial A_1(w) = \partial A_1(g(w'))$. Then by (24), we have $A_2(w) = A_2(g(w'))$. (Note that it is not difficult to prove Fact 1 mechanically via the definitions of $A_1(\cdot)$, $A_2(\cdot)$. But we omit it here.)

One can also verify Fact 2 in Fig. 5. We now give a more formal proof. If $Q(w, w') > 0$, then w and w' satisfy conditions (i)~(iii). We first show that $Q(g(w'), g(w)) > 0$. To this end, we need to verify that the states $g(w')$ and $g(w)$ satisfy condition (i)~(iii). Condition (i) holds because $\bar{A}_1(w) = \bar{A}_1(g(w'))$ by Fact 1, and because $g(\cdot)$ does not change the “on-off state” of its argument. Condition (ii) holds since $g(\cdot)$ has reversed the remaining time (cf. (29)). Condition (iii) requires that in the reversed process, any link $k \in A_2(g(w'))$ which is transmitting in the state $g(w)$ must have just started its transmission. This is true because $A_2(g(w')) = A_2(w)$ by Fact 1, and that in the original process $w(t)$, any link $k \in A_2(w)$ which is transmitting in state w must be in its last slot of the transmission (otherwise the link would be in $A_1(w)$). Then condition (iii) holds since $g(\cdot)$ has reversed the remaining time.

This completes the proof that $Q(w, w') > 0 \Rightarrow Q(g(w'), g(w)) > 0$. Now, if $Q(g(w'), g(w)) > 0$, by the above result, we have $Q(g(g(w)), g(g(w'))) > 0$. Since $g(g(w)) = w$, $g(g(w')) = w'$, we have $Q(w, w') > 0$. This completes the proof of Fact 2.

Consider two states w and w' with $Q(w, w') > 0$. Then $Q(g(w'), g(w)) > 0$, with

$$Q(g(w'), g(w)) = \prod_{i \in A_2(g(w'))} [p_i \cdot f(b_i, i, x)]^{x_i} q_i^{1-x_i}$$

by (25). Using (26) and $A_2(w) = A_2(g(w'))$, we have

$$\begin{aligned} \tilde{Q}(w', w) &= \prod_{i \in A_2(w)} [p_i \cdot f(b_i, i, x)]^{x_i} q_i^{1-x_i} \\ &= \prod_{i \in A_2(g(w'))} [p_i \cdot f(b_i, i, x)]^{x_i} q_i^{1-x_i} \\ &= Q(g(w'), g(w)). \end{aligned} \quad (30)$$

Therefore, $\tilde{Q}(w', w)$ is the transition probability of the reversed process.

By definition, $\tilde{Q}(w', w) = 0$ for any w, w' satisfying $Q(w, w') = 0$. So, given w' ,

$$\begin{aligned} \sum_w \tilde{Q}(w', w) &= \sum_{w: Q(w, w') > 0} \tilde{Q}(w', w) \\ &= \sum_{w: Q(w, w') > 0} Q(g(w'), g(w)) \\ &= \sum_{w: Q(g(w'), g(w)) > 0} Q(g(w'), g(w)) \\ &= \sum_w Q(g(w'), g(w)) = 1 \end{aligned}$$

where the last step has used the fact that $g(\cdot)$ is a one-one mapping, so that the summation is over all valid states. ■

Using Lemma 1, the probability of any on-off state x , as in Theorem 1, can be computed by summing up the probabilities of all states w 's with the same on-off state x , using (22).

Define the set of valid states $\mathcal{B}(x) := \{w \mid \text{the on-off state is } x \text{ in the state } w\}$. By Lemma 1, we have

$$\begin{aligned} p(x) &= \sum_{w \in \mathcal{B}(x)} p(w) \\ &= \frac{1}{E} \sum_{w \in \mathcal{B}(x)} \left\{ \prod_{i: x_i=0} q_i \prod_{j: x_j=1} [p_j \cdot f(b_j, j, x)] \right\} \\ &= \frac{1}{E} \left(\prod_{i: x_i=0} q_i \prod_{j: x_j=1} p_j \right) \sum_{w \in \mathcal{B}(x)} \prod_{j: x_j=1} f(b_j, j, x) \\ &= \frac{1}{E} \left(\prod_{i: x_i=0} q_i \prod_{j: x_j=1} p_j \right) \cdot \sum_{w \in \mathcal{B}(x)} \left[\prod_{j \in S(x)} P_j(b_j) \right]. \end{aligned} \quad (31)$$

Now we compute the term $\sum_{w \in \mathcal{B}(x)} \left[\prod_{j \in S(x)} P_j(b_j) \right]$. Consider a state $w = \{x, ((b_k, a_k), \forall k : x_k = 1)\} \in \mathcal{B}(x)$. For $k \in S(x)$, b_k can take different values in \mathcal{Z}_{++} . For each fixed b_k , a_k can be any integer from 1 to b_k . For a collision component $C_m(x)$ (i.e., $|C_m(x)| > 1$), the remaining time of each link in the component, $a^{(m)}$, can be any integer from 1 to γ . Then we have

$$\begin{aligned} &\sum_{w \in \mathcal{B}(x)} \left[\prod_{j \in S(x)} P_j(b_j) \right] \\ &= \prod_{j \in S(x)} \left[\sum_{b_j} \sum_{1 \leq a_j \leq b_j} P_j(b_j) \right] \prod_{m: |C_m(x)| > 1} \left(\sum_{1 \leq a^{(m)} \leq \gamma} 1 \right) \\ &= \prod_{j \in S(x)} \left[\sum_{b_j} b_j P_j(b_j) \right] \cdot \gamma^{h(x)} \\ &= \left(\prod_{j \in S(x)} T_j \right) \gamma^{h(x)}. \end{aligned} \quad (32)$$

Combining (31) and (32) completes the proof.

B. Proof of Theorem 2

1) *Some definitions:* If at an on-off state x , $k \in S(x)$ (i.e., k is transmitting successfully), it is possible that link k is transmitting the overhead or the payload. So we define the “detailed state” (x, z) , where $z \in \{0, 1\}^K$. Let $z_k = 1$ if $k \in S(x)$ and link k is transmitting its payload (instead of

overhead). Let $z_k = 0$ otherwise. Denote the set of all possible detailed states (x, z) by \mathcal{S} .

Then similar to the proof of Theorem 1, and using equation (7), we have the following product-form stationary distribution

$$p((x, z); \mathbf{r}) = \frac{1}{E(\mathbf{r})} g(x, z) \cdot \exp\left(\sum_k z_k r_k\right) \quad (33)$$

where

$$g(x, z) = g(x) \cdot (\tau')^{|S(x)| - \mathbf{1}' \mathbf{z}} T_0^{\mathbf{1}' \mathbf{z}} \quad (34)$$

where $\mathbf{1}' \mathbf{z}$ is the number of links that are transmitting the payload in state (x, z) .

Clearly, this provides another expression of the service rate $s_k(\mathbf{r})$:

$$s_k(\mathbf{r}) = \sum_{(x, z) \in \mathcal{S}: z_k=1} p((x, z); \mathbf{r}). \quad (35)$$

Now we give alternative definitions of feasible and strictly feasible arrival rates to facilitate our proof. We will show that these definitions are equivalent to Definition 1.

Definition 2: (i) A vector of arrival rate $\lambda \in \mathcal{R}_+^K$ (where K is the number of links) is *feasible* if there exists a probability distribution $\bar{\mathbf{p}}$ over \mathcal{S} (i.e., $\sum_{(x, z) \in \mathcal{S}} \bar{p}((x, z)) = 1$ and $\bar{p}((x, z)) \geq 0$), such that

$$\lambda_k = \sum_{(x, z) \in \mathcal{S}} \bar{p}((x, z)) \cdot z_k. \quad (36)$$

Let $\bar{\mathcal{C}}_{CO}$ be the set of feasible λ , where ‘‘CO’’ stands for ‘‘collision’’.

The rationale of the definition is that if λ can be scheduled by the network, the fraction of time that the network spent in the detailed states must be non-negative and sum up to 1. (Note that (36) is the probability that link k is sending its payload given the distribution of the detailed states.)

(ii) A vector of arrival rate $\lambda \in \mathcal{R}_+^K$ is *strictly feasible* if it can be written as (36) where $\sum_{(x, z) \in \mathcal{S}} \bar{p}((x, z)) = 1$ and $\bar{p}((x, z)) > 0$. Let \mathcal{C}_{CO} be the set of strictly feasible λ .

For example, in the ad-hoc network in Fig. 2 (b), $\lambda = (0.5, 0.5, 0.5)$ is feasible, because (36) holds if we let the probability of the detailed state $(x = (1, 0, 1), z = (1, 0, 1))$ be 0.5, the probability of the detailed state $(x = (0, 1, 0), z = (0, 1, 0))$ be 0.5, and all other detailed states have probability 0. However, $\lambda = (0.5, 0.5, 0.5)$ is not strictly feasible since it cannot be written as (36) where all $\bar{p}((x, z)) > 0$. But $\lambda' = (0.49, 0.49, 0.49)$ is strictly feasible.

Proposition 2: The above definitions are equivalent to Definition 1. That is,

$$\bar{\mathcal{C}}_{CO} = \bar{\mathcal{C}} \quad (37)$$

$$\mathcal{C}_{CO} = \mathcal{C}. \quad (38)$$

Proof: We first prove (37). By definition, any $\lambda \in \bar{\mathcal{C}}$ can be written as $\lambda = \sum_{\sigma \in \mathcal{X}} \bar{p}_\sigma \sigma$ where \mathcal{X} is the set of independent sets, and $\bar{\mathbf{p}} = (\bar{p}_\sigma)_{\sigma \in \mathcal{X}}$ is a probability distribution, i.e., $\bar{p}_\sigma \geq 0$, $\sum_{\sigma \in \mathcal{X}} \bar{p}_\sigma = 1$. Now we construct a distribution \mathbf{p} over the states $(x, z) \in \mathcal{S}$ as follows. Let $p((\sigma, \sigma)) = \bar{p}_\sigma, \forall \sigma \in \mathcal{X}$, and let $p((x, z)) = 0$ for all other states $(x, z) \in \mathcal{S}$. Then, clearly $\sum_{(x, z) \in \mathcal{S}} p((x, z)) \cdot z =$

$\sum_{\sigma \in \mathcal{X}} p((\sigma, \sigma)) \cdot \sigma = \sum_{\sigma \in \mathcal{X}} \bar{p}_\sigma \sigma = \lambda$, which implies that $\lambda \in \bar{\mathcal{C}}_{CO}$. So,

$$\bar{\mathcal{C}} \subseteq \bar{\mathcal{C}}_{CO}. \quad (39)$$

On the other hand, if $\lambda \in \bar{\mathcal{C}}_{CO}$, then $\lambda = \sum_{(x, z) \in \mathcal{S}} p((x, z)) \cdot z$ for some distribution \mathbf{p} over \mathcal{S} . We define another distribution $\bar{\mathbf{p}}$ over \mathcal{X} as follows. Let $\bar{p}_\sigma = \sum_{(x, z) \in \mathcal{S}: z=\sigma} p((x, z)), \forall \sigma \in \mathcal{X}$. Then, $\lambda = \sum_{(x, z) \in \mathcal{S}} p((x, z)) \cdot z = \sum_{\sigma \in \mathcal{X}} \sum_{(x, z) \in \mathcal{S}: z=\sigma} p((x, z)) \sigma = \sum_{\sigma \in \mathcal{X}} \bar{p}_\sigma \sigma$, which implies that $\lambda \in \bar{\mathcal{C}}$. Therefore

$$\bar{\mathcal{C}}_{CO} \subseteq \bar{\mathcal{C}}. \quad (40)$$

Combining (39) and (40) yields (37).

We defined that \mathcal{C} is the interior of $\bar{\mathcal{C}}$. To prove (38), we only need to show that \mathcal{C}_{CO} is also the interior of $\bar{\mathcal{C}}$. The proof is similar to that in Appendix A of [32], and is thus omitted. ■

2) *Existence of \mathbf{r}^** : Assume that λ is strictly feasible. Consider the following convex optimization problem, where the vector \mathbf{u} can be viewed as a probability distribution over the detailed states (x, z) :

$$\begin{aligned} \max_{\mathbf{u}} \quad & \{H(\mathbf{u}) + \sum_{(x, z) \in \mathcal{S}} [u_{(x, z)} \cdot \log(g(x, z))]\} \\ \text{s.t.} \quad & \sum_{(x, z) \in \mathcal{S}: z_k=1} u_{(x, z)} = \lambda_k, \forall k \\ & u_{(x, z)} \geq 0, \sum_{(x, z) \in \mathcal{S}} u_{(x, z)} = 1 \end{aligned} \quad (41)$$

where $H(\mathbf{u}) := \sum_{(x, z) \in \mathcal{S}} [-u_{(x, z)} \log(u_{(x, z)})]$ is the ‘‘entropy’’ of the distribution \mathbf{u} .

Let r_k be the dual variable associated with the constraint $\sum_{(x, z) \in \mathcal{S}: z_k=1} u_{(x, z)} = \lambda_k$, and let the vector $\mathbf{r} := (r_k)$. We will show the following.

Lemma 2: The optimum dual variables \mathbf{r}^* (when problem (41) is solved) exists, and satisfy (10), i.e., $s_k(\mathbf{r}^*) = \lambda_k, \forall k$. Also, the dual problem of (41) is (11).

Proof: With the above definition of \mathbf{r} , a partial Lagrangian of problem (41) (subject to $u_{(x, z)} \geq 0, \sum_{(x, z) \in \mathcal{S}} u_{(x, z)} = 1$) is

$$\begin{aligned} \mathcal{L}(\mathbf{u}; \mathbf{r}) &= H(\mathbf{u}) + \sum_{(x, z) \in \mathcal{S}} [u_{(x, z)} \log(g(x, z))] \\ &+ \sum_k r_k \left[\sum_{(x, z) \in \mathcal{S}: z_k=1} u_{(x, z)} - \lambda_k \right] \\ &= \sum_{(x, z) \in \mathcal{S}} \{u_{(x, z)} [-\log(u_{(x, z)}) + \log(g(x, z))] \\ &+ \sum_{k: z_k=1} r_k\} - \sum_k (r_k \lambda_k). \end{aligned} \quad (42)$$

So

$$\frac{\partial \mathcal{L}(\mathbf{u}; \mathbf{r})}{\partial u_{(x, z)}} = -\log(u_{(x, z)}) - 1 + \log(g(x, z)) + \sum_{k: z_k=1} r_k.$$

We claim that

$$u_{(x, z)}(\mathbf{r}) := p((x, z); \mathbf{r}), \forall (x, z) \in \mathcal{S} \quad (43)$$

(cf. equation (33)) maximizes $\mathcal{L}(\mathbf{u}; \mathbf{r})$ over \mathbf{u} subject to $u_{(x,z)} \geq 0$, $\sum_{(x,z) \in \mathcal{S}} u_{(x,z)} = 1$. Indeed, the partial derivative at the point $\mathbf{u}(\mathbf{r})$ is

$$\frac{\partial \mathcal{L}(\mathbf{u}(\mathbf{r}); \mathbf{r})}{\partial u_{(x,z)}} = \log(E(\mathbf{r})) - 1,$$

which is the same for all $(x, z) \in \mathcal{S}$ (Since given the dual variables \mathbf{r} , $\log(E(\mathbf{r}))$ is a constant). Also, $u_{(x,z)}(\mathbf{r}) = p((x, z); \mathbf{r}) > 0$ and $\sum_{(x,z) \in \mathcal{S}} u_{(x,z)}(\mathbf{r}) = 1$. Therefore, it is impossible to increase $\mathcal{L}(\mathbf{u}; \mathbf{r})$ by slightly perturbing \mathbf{u} around $\mathbf{u}(\mathbf{r})$ (subject to $\mathbf{1}^T \mathbf{u} = 1$). Since $\mathcal{L}(\mathbf{u}; \mathbf{r})$ is concave in \mathbf{u} , the claim follows.

Denote $l(\mathbf{r}) = \max_{\mathbf{u}} \mathcal{L}(\mathbf{u}; \mathbf{r})$, then the dual problem of (41) is $\inf_{\mathbf{r}} l(\mathbf{r})$. Plugging the expression of $u_{(x,z)}(\mathbf{r})$ into $\mathcal{L}(\mathbf{u}; \mathbf{r})$, it is not difficult to find that $\inf_{\mathbf{r}} l(\mathbf{r})$ is equivalent to $\sup_{\mathbf{r}} L(\mathbf{r}; \lambda)$ where $L(\mathbf{r}; \lambda)$ is defined in (12).

Since λ is strictly feasible, it can be written as (36) where $\sum_{(x,z) \in \mathcal{S}} \bar{p}((x, z)) = 1$ and $\bar{p}((x, z)) > 0$. Therefore, there exists $\mathbf{u} \succ \mathbf{0}$ (by choosing $\mathbf{u} = \bar{\mathbf{p}}$) that satisfies the constraints in (41) and also in the interior of the domain of the objective function. So, problem (41) satisfies the Slater condition [2]. As a result, there exists a vector of (finite) optimal dual variables \mathbf{r}^* when problem (41) is solved. Also, \mathbf{r}^* solves the dual problem $\sup_{\mathbf{r}} L(\mathbf{r}; \lambda)$. Therefore, $\sup_{\mathbf{r}} L(\mathbf{r}; \lambda)$ is attainable and can be written as $\max_{\mathbf{r}} L(\mathbf{r}; \lambda)$, as in (11).

Finally, the optimal solution \mathbf{u}^* of problem (41) is such that $u_{(x,z)}^* = u_{(x,z)}(\mathbf{r}^*)$, $\forall (x, z) \in \mathcal{S}$. Also, \mathbf{u}^* is clearly feasible for problem (41). Therefore,

$$\sum_{(x,z) \in \mathcal{S}: z_k=1} u_{(x,z)}^* = s_k(\mathbf{r}^*) = \lambda_k, \forall k.$$

Remark: From (42) and (43), we see that a subgradient (or gradient) of the dual objective function $L(\mathbf{r}; \lambda)$ is

$$\frac{\partial L(\mathbf{r}; \lambda)}{\partial r_k} = \lambda_k - \sum_{(x,z) \in \mathcal{S}: z_k=1} u_{(x,z)}(\mathbf{r}) = \lambda_k - s_k(\mathbf{r}).$$

This can also be obtained by direct differentiation of $L(\mathbf{r}; \lambda)$.

3) *Uniqueness of \mathbf{r}^* :* Now we show the uniqueness of \mathbf{r}^* . Note that the objective function of (41) is strictly concave. Therefore \mathbf{u}^* , the optimal solution of (41) is unique. Consider two detailed state $(\mathbf{e}_k, \mathbf{e}_k)$ and $(\mathbf{e}_k, \mathbf{0})$, where \mathbf{e}_k is the K -dimensional vector whose k 'th element is 1 and all other elements are 0's. We have $u_{(\mathbf{e}_k, \mathbf{e}_k)}^* = p((\mathbf{e}_k, \mathbf{e}_k); \mathbf{r}^*)$ and $u_{(\mathbf{e}_k, \mathbf{0})}^* = p((\mathbf{e}_k, \mathbf{0}); \mathbf{r}^*)$. Then by (33),

$$u_{(\mathbf{e}_k, \mathbf{e}_k)}(\mathbf{r}^*)/u_{(\mathbf{e}_k, \mathbf{0})}(\mathbf{r}^*) = \exp(r_k^*) \cdot (T_0/\tau'). \quad (44)$$

Suppose that \mathbf{r}^* is not unique, that is, there exist $\mathbf{r}_I^* \neq \mathbf{r}_{II}^*$ but both are optimal \mathbf{r} . Then, $r_{I,k}^* \neq r_{II,k}^*$ for some k . This contradicts (44) and the uniqueness of \mathbf{u}^* . Therefore \mathbf{r}^* is unique. This also implies that $\max_{\mathbf{r}} L(\mathbf{r}; \lambda)$ has a unique solution \mathbf{r}^* .

C. Proof of Theorem 3

We will use results in [23] to prove Theorem 3. Similar techniques have been used in [22] to analyze the convergence of an algorithm in [16].

1) *Part (i): Decreasing step size:* Define the concave function

$$\tilde{H}(y) := \begin{cases} -(r_{\min} - y)^2/2 & \text{if } y < r_{\min} \\ 0 & \text{if } y \in [r_{\min}, r_{\max}] \\ -(r_{\max} - y)^2/2 & \text{if } y > r_{\max}. \end{cases} \quad (45)$$

Note that $d\tilde{H}(y)/dy = \tilde{h}(y)$ where $\tilde{h}(y)$ is defined in (14). Let $G(\mathbf{r}; \lambda) := L(\mathbf{r}; \lambda) + \sum_k \tilde{H}(r_k)$. Since λ is strictly feasible, $\max_{\mathbf{r}} L(\mathbf{r}; \lambda)$ has a unique solution \mathbf{r}^* . That is, $L(\mathbf{r}^*; \lambda) > L(\mathbf{r}; \lambda)$, $\forall \mathbf{r} \neq \mathbf{r}^*$. Since $\mathbf{r}^* \in (r_{\min}, r_{\max})^K$ by assumption, then $\forall \mathbf{r}$, $\sum_k \tilde{H}(r_k^*) = 0 \geq \sum_k \tilde{H}(r_k)$. Therefore, $G(\mathbf{r}^*; \lambda) > G(\mathbf{r}; \lambda)$, $\forall \mathbf{r} \neq \mathbf{r}^*$. So \mathbf{r}^* is the unique solution of $\max_{\mathbf{r}} G(\mathbf{r}; \lambda)$. Because $\partial G(\mathbf{r}; \lambda)/\partial r_k = \lambda_k - s_k(\mathbf{r}) + \tilde{h}(r_k)$, Algorithm 1 tries to solve $\max_{\mathbf{r}} G(\mathbf{r}; \lambda)$ with inaccurate gradients.

Let $\mathbf{v}^s(t)$ be the solution of the following differential equation (for $t \geq s$)

$$dv_k(t)/dt = \lambda_k - s_k(\mathbf{v}(t)) + \tilde{h}(v_k(t)), \forall k \quad (46)$$

with the initial condition that $\mathbf{v}^s(s) = \bar{\mathbf{r}}(s)$. So, $\mathbf{v}^s(t)$ can be viewed as the “ideal” trajectory of Algorithm 1 with the smoothed arrival rate and service rate. And (46) can be viewed as a continuous-time gradient algorithm to solve $\max_{\mathbf{r}} G(\mathbf{r}; \lambda)$. We have shown above that \mathbf{r}^* is the unique solution of $\max_{\mathbf{r}} G(\mathbf{r}; \lambda)$. Therefore $\mathbf{v}^s(t)$ converges to the unique \mathbf{r}^* for any initial condition.

Recall that in Algorithm 1, $\mathbf{r}(i)$ is always updated at the beginning of a minislot. Define $Y(i-1) := (s'_k(i), w_0(i))$ where $w_0(i)$ is the state w at time t_i . Then $\{Y(i)\}$ is a non-homogeneous Markov process whose transition kernel from time t_{i-1} to t_i depends on $\mathbf{r}(i-1)$. The update in Algorithm 1 can be written as

$$r_k(i) = r_k(i-1) + \alpha(i) \cdot [f(r_k(i-1), Y(i-1)) + M(i)]$$

where $f(r_k(i-1), Y(i-1)) := \lambda_k - s'_k(i) + \tilde{h}(r_k(i-1))$, and $M(i) = \lambda'_k(i) - \lambda_k$ is zero-mean noise.

To use Corollary 8 in page 74 of [23] to show Algorithm 1's almost-sure convergence to \mathbf{r}^* , the following conditions are sufficient:

(i) $f(\cdot, \cdot)$ is Lipschitz in the first argument, and uniformly in the second argument. This holds by the construction of $\tilde{h}(\cdot)$;
(ii) The transition kernel of $Y(i)$ is continuous in $\mathbf{r}(i)$. This is true due to the way we randomize the transmission lengths in (15).

(iii) (46) has a unique convergent point \mathbf{r}^* , which has been shown above;

(iv) With Algorithm 1, $r_k(i)$ is bounded $\forall k, i$ almost surely. This is proved in Lemma 3 below.

(v) Tightness condition ((†) in [23], page 71): This is satisfied since $Y(i)$ has a bounded state-space (cf. conditions (6.4.1) and (6.4.2) in [23], page 76). The state space of $Y(i)$ is bounded because $s'_k(i) \in [0, 1]$ and $w_0(i)$ is in a finite set (which is shown in Lemma 4) below.

So, by [23], $\mathbf{r}(i)$ converges to \mathbf{r}^* almost surely.

Lemma 3: With Algorithm 1, $\mathbf{r}(i)$ is always bounded. Specifically, $r_k(i) \in [r_{\min} - 2, r_{\max} + 2\bar{\lambda}]$, $\forall k, i$, where $\bar{\lambda}$,

as defined before, is the maximal instantaneous arrival rate, so that $\lambda'_k(i) \leq \bar{\lambda}, \forall k, i$.

Proof: We first prove the upper bound $r_{max} + 2\bar{\lambda}$ by induction: (a) $r_k(0) \leq r_{max} \leq r_{max} + 2\bar{\lambda}$; (b) For $i \geq 1$, if $r_k(i-1) \in [r_{max} + \bar{\lambda}, r_{max} + 2\bar{\lambda}]$, then $\tilde{h}(r_k(i-1)) \leq -\bar{\lambda}$. Since $\lambda'_k(i) - s'_k(i) \leq \bar{\lambda}$, we have $r_k(i) \leq r_k(i-1) \leq r_{max} + 2\bar{\lambda}$. If $r_k(i-1) \in (r_{min}, r_{max} + \bar{\lambda})$, then $\tilde{h}(r_k(i-1)) \leq 0$. Also since $\lambda'_k(i) - s'_k(i) \leq \bar{\lambda}$ and $\alpha(i) \leq 1, \forall i$, we have $r_k(i) \leq r_k(i-1) + \bar{\lambda} \cdot \alpha(i) \leq r_{max} + 2\bar{\lambda}$. If $r_k(i-1) \leq r_{min}$, then

$$\begin{aligned} r_k(i) &= r_k(i-1) + \alpha(i)[\lambda'_k(i) - s'_k(i) + \tilde{h}(r_k(i-1))] \\ &\leq r_k(i-1) + \alpha(i)\{\bar{\lambda} + [r_{min} - r_k(i-1)]\} \\ &= [1 - \alpha(i)] \cdot r_k(i-1) + \alpha(i)\{\bar{\lambda} + r_{min}\} \\ &\leq [1 - \alpha(i)] \cdot r_{min} + \alpha(i)\{\bar{\lambda} + r_{min}\} \\ &= r_{min} + \alpha(i) \cdot \bar{\lambda} \\ &\leq \bar{\lambda} + r_{min} \leq r_{max} + 2\bar{\lambda}. \end{aligned}$$

The lower bound $r_{min} - 2$ can be proved similarly. ■

Lemma 4: In Algorithm 1, $w_0(i)$ is in a finite set.

Proof: By Lemma 3, we know that $r_k(i) \leq r_{max} + 2\bar{\lambda}, \forall k, i$, so $T_k^p(i) \leq T_0 \exp(r_{max} + 2\bar{\lambda}), \forall k, i$. By (15), we have $\tau_k^p(i) \leq T_0 \exp(r_{max} + 2\bar{\lambda}) + 1, \forall k, i$. Therefore, in state $w_0(i) = \{x, ((b_k, a_k), \forall k : x_k = 1)\}$, we have $b_k \leq b_{max}$ for a constant b_{max} and $a_k \leq b_k$ for any k such that $x_k = 1$. So, $w_0(i)$ is in a finite set. ■

2) *Part (ii): Constant step size:* The intuition is the same as part (i). That is, if the constant step size is small enough, then the algorithm approximately solves problem $\max_{\mathbf{r}} G(\mathbf{r}; \lambda)$. Please refer to [30] for the full proof.

D. Proof of Theorem 4

Let N' be the number of detailed states (x, z) 's, and $\mathbf{u} \in \mathcal{R}_+^{N'}$ be a probability distribution over the detailed states. For convenience of notation, we use $i = 1, 2, \dots, N'$ to index the detailed states. Then u_i , the i -th element of \mathbf{u} , is the probability of the i -th detailed state. Let \mathcal{P} be the set of N' -dimensional probability distributions, i.e., $\mathcal{P} := \{\mathbf{u}' \in \mathcal{R}_+^{N'} | \mathbf{1}^T \mathbf{u}' = 1\}$. Also define a $K \times N'$ matrix A where the element $A_{k,i} = 1$ if link k is transmitting its *payload* in the i -th detailed state, and $A_{k,i} = 0$ otherwise. Then, $A \cdot \mathbf{u}$ is the vector of achieved throughputs of the K links under the distribution \mathbf{u} .

Lemma 5: Given $\bar{\lambda}$ at the boundary of $\bar{\mathcal{C}}$, there exists a constant $b > 0$ such that the following holds: For any $0 < \epsilon < 1$, if $\mathbf{u} \in \mathcal{P}$ and $A \cdot \mathbf{u} = (1 - \epsilon)\bar{\lambda}$, then one can find $\bar{\mathbf{u}} \in \mathcal{P}$ such that $A \cdot \bar{\mathbf{u}} = \bar{\lambda}$ and $\|\mathbf{u} - \bar{\mathbf{u}}\|_1 \leq 2b \cdot \epsilon$.

Proof: Consider the set $\mathcal{U} := \{\bar{\mathbf{u}} | \bar{\mathbf{u}} \in \mathcal{P}, A \cdot \bar{\mathbf{u}} = \bar{\lambda}\}$. Clearly \mathcal{U} is a polytope since \mathcal{U} is defined by linear constraints. Denote by $\mathcal{E}(\mathcal{U})$ the set of extreme points of \mathcal{U} . Then, if $\mathbf{u} \in \mathcal{U}$ and satisfies $A \cdot \mathbf{u} = (1 - \epsilon)\bar{\lambda}$, we have

$$\mathbf{u} = \sum_{\mathbf{y} \in \mathcal{E}(\mathcal{U})} a_{\mathbf{y}} \mathbf{y} \quad (47)$$

where $a_{\mathbf{y}} \geq 0$ and $\sum_{\mathbf{y} \in \mathcal{E}(\mathcal{U})} a_{\mathbf{y}} = 1$. Also, by definition, for any $\mathbf{y} \in \mathcal{E}(\mathcal{U})$, $A \cdot \mathbf{y} = \rho_{\mathbf{y}} \bar{\lambda}$ for some $\rho_{\mathbf{y}} \in [0, 1]$. Then,

$$A \cdot \mathbf{u} = \sum_{\mathbf{y} \in \mathcal{E}(\mathcal{U})} (a_{\mathbf{y}} A \cdot \mathbf{y}) = \rho_{\mathbf{u}} \bar{\lambda}$$

where $\rho_{\mathbf{u}} = \sum_{\mathbf{y} \in \mathcal{E}(\mathcal{U})} (a_{\mathbf{y}} \rho_{\mathbf{y}}) = 1 - \epsilon$.

For any $\mathbf{u}' \in \mathcal{U}$, define

$$\begin{aligned} D(\mathbf{u}') &:= \min_{\mathbf{z}} \|\mathbf{u}' - \mathbf{z}\|_1 \\ \text{s.t. } &A \cdot \mathbf{z} = \bar{\lambda}. \end{aligned} \quad (48)$$

It can be shown that $D(\mathbf{u}')$ is a convex function of \mathbf{u}' . (Proof: Consider any two \mathbf{u}'_I and \mathbf{u}'_{II} . When $\mathbf{u}' = \mathbf{u}'_I$ (or \mathbf{u}'_{II}), suppose \mathbf{z}_I (or \mathbf{z}_{II}) is an optimal solution of problem (48). That is, $D(\mathbf{u}'_I) = \|\mathbf{u}'_I - \mathbf{z}_I\|_1$ and $D(\mathbf{u}'_{II}) = \|\mathbf{u}'_{II} - \mathbf{z}_{II}\|_1$. Given any $\beta \in [0, 1]$, we have

$$\begin{aligned} &\beta \cdot D(\mathbf{u}'_I) + (1 - \beta) \cdot D(\mathbf{u}'_{II}) \\ &= \beta \|\mathbf{u}'_I - \mathbf{z}_I\|_1 + (1 - \beta) \|\mathbf{u}'_{II} - \mathbf{z}_{II}\|_1 \\ &\geq \|\beta(\mathbf{u}'_I - \mathbf{z}_I) + (1 - \beta)(\mathbf{u}'_{II} - \mathbf{z}_{II})\|_1 \\ &= \|[\beta \mathbf{u}'_I + (1 - \beta)\mathbf{u}'_{II}] - [\beta \mathbf{z}_I + (1 - \beta)\mathbf{z}_{II}]\|_1. \end{aligned} \quad (49)$$

Now consider $D(\beta \mathbf{u}'_I + (1 - \beta)\mathbf{u}'_{II})$. Clearly, $\tilde{\mathbf{z}} := \beta \mathbf{z}_I + (1 - \beta)\mathbf{z}_{II}$ satisfies the constraint $A \cdot \tilde{\mathbf{z}} = \bar{\lambda}$. So, $D(\beta \mathbf{u}'_I + (1 - \beta)\mathbf{u}'_{II}) \leq \|[\beta \mathbf{u}'_I + (1 - \beta)\mathbf{u}'_{II}] - [\beta \mathbf{z}_I + (1 - \beta)\mathbf{z}_{II}]\|_1$. Combined with (49), we have $D(\beta \mathbf{u}'_I + (1 - \beta)\mathbf{u}'_{II}) \leq \beta \cdot D(\mathbf{u}'_I) + (1 - \beta) \cdot D(\mathbf{u}'_{II})$, which implies that $D(\mathbf{u}')$ is a convex function of \mathbf{u}' .

Using this fact and (47), we have

$$D(\mathbf{u}) \leq \sum_{\mathbf{y} \in \mathcal{E}(\mathcal{U})} (a_{\mathbf{y}} D(\mathbf{y})) = \sum_{\mathbf{y} \in \mathcal{E}(\mathcal{U}), \rho_{\mathbf{y}} < 1} (a_{\mathbf{y}} D(\mathbf{y})) \quad (50)$$

where the second step is because if $\rho_{\mathbf{y}} = 1$, then $D(\mathbf{y}) = 0$.

Define

$$b := \frac{1}{2} \max_{\mathbf{y} \in \mathcal{E}(\mathcal{U}), \rho_{\mathbf{y}} < 1} D(\mathbf{y}) / (1 - \rho_{\mathbf{y}}). \quad (51)$$

Since there are a finite number of elements in $\mathcal{E}(\mathcal{U})$, we have $0 < b < +\infty$. Then

$$\begin{aligned} D(\mathbf{u}) &\leq \sum_{\mathbf{y} \in \mathcal{E}(\mathcal{U}), \rho_{\mathbf{y}} < 1} (a_{\mathbf{y}} D(\mathbf{y})) \\ &\leq 2b \cdot \sum_{\mathbf{y} \in \mathcal{E}(\mathcal{U}), \rho_{\mathbf{y}} < 1} [a_{\mathbf{y}} \cdot (1 - \rho_{\mathbf{y}})] \\ &= 2b \cdot \sum_{\mathbf{y} \in \mathcal{E}(\mathcal{U})} [a_{\mathbf{y}} \cdot (1 - \rho_{\mathbf{y}})] \\ &= 2b[1 - \rho_{\mathbf{u}}] = 2b\epsilon. \end{aligned}$$

Let $\bar{\mathbf{u}}$ be a solution of (48) with $\mathbf{u}' = \mathbf{u}$, then $\|\mathbf{u} - \bar{\mathbf{u}}\|_1 = D(\mathbf{u}) \leq 2b\epsilon$. ■

Lemma 6: Assume that $\mathbf{u}, \bar{\mathbf{u}} \in \mathcal{P}$ and $\mathbf{u} \neq \bar{\mathbf{u}}$. If $\|\mathbf{u} - \bar{\mathbf{u}}\|_1 \leq 2c$ for a constant $c \in (0, 1]$, then $|H(\mathbf{u}) - H(\bar{\mathbf{u}})| \leq c \cdot [\log(N'/c) + 1]$, where $H(\mathbf{u}) := \sum_{i=1}^{N'} [-u_i \log(u_i)]$ is the "entropy" of the distribution \mathbf{u} .

Proof: Let $(\mathbf{u} - \bar{\mathbf{u}})^+$ and $(\mathbf{u} - \bar{\mathbf{u}})^-$ be the positive part and negative part of $\mathbf{u} - \bar{\mathbf{u}}$. Then clearly $\|\mathbf{u} - \bar{\mathbf{u}}\|_1 = \|(\mathbf{u} - \bar{\mathbf{u}})^+ + (\mathbf{u} - \bar{\mathbf{u}})^-\|_1$.

$\bar{\mathbf{u}}^+\|_1 + \|(\mathbf{u} - \bar{\mathbf{u}})^-\|_1$. Also, since $\mathbf{1}^T \mathbf{u} = \mathbf{1}^T \bar{\mathbf{u}} = 1$, we have $\|(\mathbf{u} - \bar{\mathbf{u}})^+\|_1 = \|(\mathbf{u} - \bar{\mathbf{u}})^-\|_1$. Therefore,

$$0 < \|(\mathbf{u} - \bar{\mathbf{u}})^+\|_1 = \|(\mathbf{u} - \bar{\mathbf{u}})^-\|_1 = \frac{1}{2} \|\mathbf{u} - \bar{\mathbf{u}}\|_1 \leq c.$$

Let the index sets $\mathcal{I}_1 = \{i | i \in \{1, 2, \dots, N'\}, u_i > \bar{u}_i\}$ and $\mathcal{I}_2 = \{i | i \in \{1, 2, \dots, N'\}, u_i < \bar{u}_i\}$. Define $N_1 := |\mathcal{I}_1|, N_2 := |\mathcal{I}_2|$. Clearly $N_1 + N_2 \leq N'$, and $N_1, N_2 > 0$.

Now we show that $H(\mathbf{u}) - H(\bar{\mathbf{u}}) \leq c \cdot [\log(N'/c) + 1]$. Define the function $f(x) := x \log(1/x)$. First, for $i \in \mathcal{I}_1$, denote $\delta_i = u_i - \bar{u}_i > 0$, then $\|(\mathbf{u} - \bar{\mathbf{u}})^+\|_1 = \sum_{i \in \mathcal{I}_1} \delta_i$. Note that $u_i, \bar{u}_i, \delta_i \in [0, 1]$. Since $f(x)$ is concave, with $u_i > \bar{u}_i \geq 0$, we have $f(u_i) - f(\bar{u}_i) \leq f(\delta_i) - f(0) = f(\delta_i)$. So

$$\begin{aligned} \sum_{i \in \mathcal{I}_1} [f(u_i) - f(\bar{u}_i)] &\leq \sum_{i \in \mathcal{I}_1} f(\delta_i) \\ &\leq \sum_{i \in \mathcal{I}_1} f\left(\frac{N_1}{\|(\mathbf{u} - \bar{\mathbf{u}})^+\|_1}\right) \\ &= \|(\mathbf{u} - \bar{\mathbf{u}})^+\|_1 \log\left(\frac{N_1}{\|(\mathbf{u} - \bar{\mathbf{u}})^+\|_1}\right) \\ &\leq \|(\mathbf{u} - \bar{\mathbf{u}})^+\|_1 \log\left(\frac{N'}{\|(\mathbf{u} - \bar{\mathbf{u}})^+\|_1}\right) \end{aligned} \quad (52)$$

where the second inequality follows from the fact that $f(x)$ is concave and $\sum_{i \in \mathcal{I}_1} \delta_i = \|(\mathbf{u} - \bar{\mathbf{u}})^+\|_1$.

It is easy to show that $f(x)$ is increasing in the range $x \in [0, \exp(-1)]$. Also, we have $N' \geq 3$ (even in the smallest one-link network). Thus $\|(\mathbf{u} - \bar{\mathbf{u}})^+\|_1 / N' \leq c / N' \leq 1 / N' < \exp(-1)$. Therefore, from (52),

$$\begin{aligned} \sum_{i \in \mathcal{I}_1} [f(u_i) - f(\bar{u}_i)] &\leq N' \cdot f\left(\frac{\|(\mathbf{u} - \bar{\mathbf{u}})^+\|_1}{N'}\right) \\ &\leq N' \cdot f\left(\frac{c}{N'}\right) = c \log\left(\frac{N'}{c}\right). \end{aligned} \quad (53)$$

Since $f(x)$ is concave, we have $f(u_i) - f(\bar{u}_i) \leq (u_i - \bar{u}_i) f'(\bar{u}_i)$. For $i \in \mathcal{I}_2$, since $u_i < \bar{u}_i$ and $f'(\bar{u}_i) \geq -1$ for any $\bar{u}_i \in (0, 1]$, we have

$$\sum_{i \in \mathcal{I}_2} [f(u_i) - f(\bar{u}_i)] \leq \sum_{i \in \mathcal{I}_2} (\bar{u}_i - u_i) \leq c. \quad (54)$$

Using (53) and (54), we conclude that $H(\mathbf{u}) - H(\bar{\mathbf{u}}) \leq c \cdot [\log(N'/c) + 1]$. The same argument shows that $H(\bar{\mathbf{u}}) - H(\mathbf{u}) \leq c \cdot [\log(N'/c) + 1]$. Therefore the lemma follows. ■ Now we are ready to prove Theorem 4.

As explained before, $\mathbf{r}^*(\lambda)$ is the vector of optimal dual variables in the optimization problem (41), simply written as

$$\begin{aligned} V(\lambda) &:= \max_{\mathbf{v} \in \mathcal{P}} H(\mathbf{v}) + \sum_i g_i v_i \\ \text{s.t.} \quad &A \cdot \mathbf{v} = \lambda \end{aligned} \quad (55)$$

where $g_i, i = 1, 2, \dots, N'$ denote the constants $\log(g(x, z))$ for $(x, z) \in \mathcal{S}$. Let \mathbf{u}^* be the optimal solution when $\lambda = (1 - \epsilon)\bar{\lambda}$. Then $V((1 - \epsilon)\bar{\lambda}) = H(\mathbf{u}^*) + \sum_i g_i u_i^*$.

Consider the case when $\epsilon \leq 1/b$. By Lemma 5, there exists $\bar{\mathbf{u}} \in \mathcal{P}$ such that $A \cdot \bar{\mathbf{u}} = \bar{\lambda}$ and $\|\mathbf{u}^* - \bar{\mathbf{u}}\|_1 \leq 2b \cdot \epsilon \leq 2$. By Lemma 6,

$$|H(\mathbf{u}^*) - H(\bar{\mathbf{u}})| \leq b \cdot \epsilon [\log(\frac{N'}{b \cdot \epsilon}) + 1]. \quad (56)$$

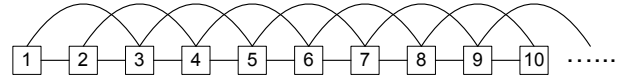


Fig. 11: Conflict graph of a line network

Since $\bar{\mathbf{u}} \in \mathcal{P}$ satisfies $A \cdot \bar{\mathbf{u}} = \bar{\lambda}$, we have $V(\bar{\lambda}) \geq H(\bar{\mathbf{u}}) + \sum_i g_i \bar{u}_i$. Also, the “value function” $V(\lambda)$ is concave in λ ([2] page 250), and satisfies $\nabla V(\lambda) = -\mathbf{r}^*(\lambda)$. Therefore

$$\begin{aligned} H(\bar{\mathbf{u}}) + \sum_i g_i \bar{u}_i &\leq V(\bar{\lambda}) \\ &\leq V((1 - \epsilon)\bar{\lambda}) + (\bar{\lambda} - (1 - \epsilon)\bar{\lambda})^T [-\mathbf{r}^*((1 - \epsilon)\bar{\lambda})] \\ &= H(\mathbf{u}^*) + \sum_i g_i u_i^* - \epsilon \cdot \bar{\lambda}^T \mathbf{r}^*((1 - \epsilon)\bar{\lambda}). \end{aligned}$$

So,

$$\epsilon \cdot \bar{\lambda}^T \mathbf{r}^*((1 - \epsilon)\bar{\lambda}) \leq H(\mathbf{u}^*) - H(\bar{\mathbf{u}}) + \sum_i g_i (u_i^* - \bar{u}_i). \quad (57)$$

Denote $G := \max_i |g_i|$, then

$$\begin{aligned} \left| \sum_i g_i (u_i^* - \bar{u}_i) \right| &\leq \sum_i (|g_i| \cdot |u_i^* - \bar{u}_i|) \\ &\leq G \sum_i |u_i^* - \bar{u}_i| \leq 2Gb\epsilon. \end{aligned} \quad (58)$$

Combining (57), (58) and (56), we have $\epsilon \cdot \bar{\lambda}^T \mathbf{r}^*((1 - \epsilon)\bar{\lambda}) \leq b \cdot \epsilon [\log(\frac{N'}{b \cdot \epsilon}) + 1] + 2Gb\epsilon$, which proves (18).

For the second case when $\epsilon > 1/b$, choose an arbitrary $\bar{\mathbf{u}} \in \mathcal{P}$ such that $A \cdot \bar{\mathbf{u}} = \bar{\lambda}$. Clearly, $\|\mathbf{u}^* - \bar{\mathbf{u}}\|_1 \leq 2$, so the RHS of (58) can be replaced by $2G$. Also, $|H(\mathbf{u}^*) - H(\bar{\mathbf{u}})| \leq \log(N')$ since $H(\cdot) \in [0, \log(N')]$. Using these in (57) yields (19).

E. Simulations of 1-D and 2-D lattice topologies

Consider a line network (i.e., 1-D lattice network) with 16 links, where each link conflicts with the nearest 2 links on each side (so, link k conflicts with 4 other links if it is not at the two ends of the network), as in Fig. 11. Let $\gamma = 5, \tau' = 10$ and $p_k = 1/16, \forall k$. In each simulation, we let all links use the same, fixed payload length $T^p = 30, 50, 100, 150$, and we compute the “short-term throughput” of link 8 (in the middle of the network) every 50 milliseconds (or $50/0.009 \approx 5556$ slots). That is, we compute the average throughput of link 8 in each time window of 50 milliseconds.

Note that a successful transmission has a length of $\tau' + T^p$. Also note that if the parameter r_{max} in Algorithm 1 satisfies that $T_0 \exp(r_{max}) = T^p$, then it is possible that all links transmit payload T^p .

The results are plotted in Fig. 12. We see that the oscillation in the short-term throughput increases as T^p increases, indicating worsening short-term fairness.

Next we simulate a 2-D lattice network with $5 \times 5 = 25$ links in Fig. 13. Each link conflicts with the nearest 4 links around it (if it's not at the boundary). Similar to the previous simulation, we use different values of T^p and obtain the following short-term throughput of link 13 (which is at the center of the network) in Fig. 14. Again we observe worsening short-term fairness as T^p increases. Also, the oscillation of link 13's short-term throughput is greater than link 8 in the line network, since

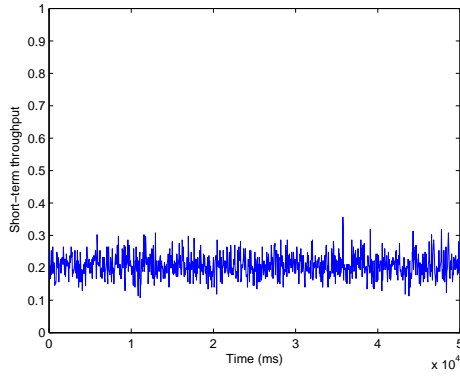
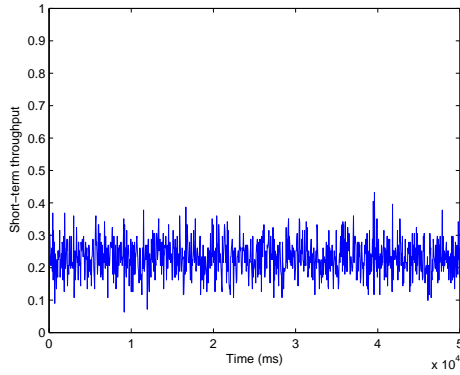
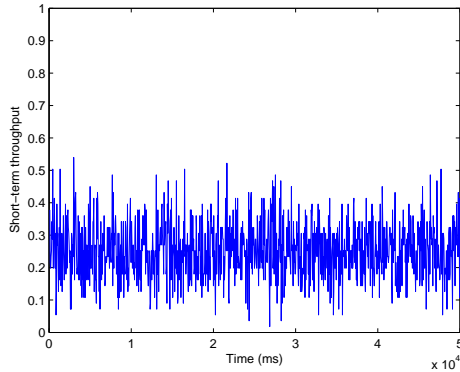
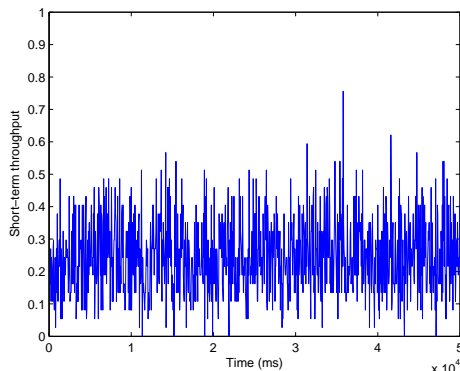
(a) $T^P = 30$ (b) $T^P = 50$ (c) $T^P = 100$ (d) $T^P = 150$

Fig. 12: Short-term throughput of link 8 in the line network

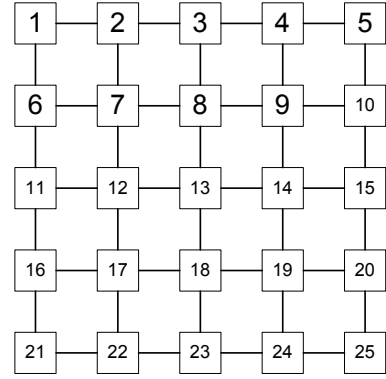


Fig. 13: Conflict graph of a 2-D lattice network

link 13 has 4 conflicting neighbors which do not conflict with each other.

However, as mentioned before, it remains an open problem to exact characterize the relationship between short-term fairness (in particular the standard deviation of the access delay) and T^P (or equivalently, r_{max}) in general topologies. We are interested to further explore useful methods to quantify the relationship.

REFERENCES

- [1] F. P. Kelly, “*Reversibility and Stochastic Networks*,” Wiley, 1979.
- [2] S. Boyd and L. Vandenberghe, “*Convex Optimization*”, Cambridge University Press, 2004.
- [3] L. Tassiulas and A. Ephremides, “Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks,” *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936-1948, Dec. 1992.
- [4] X. Lin, N.B. Shroff, R. Srikant, “A Tutorial on Cross-Layer Optimization in Wireless Networks,” *IEEE Journal on Selected Areas in Communications*, 2006.
- [5] J. M. Harrison and R. J. Williams, “Workload interpretation for Brownian models of stochastic processing networks,” *Mathematics of Operations Research*, vol. 32, pp. 808-820, 2007.
- [6] P. Chaporkar, K. Kar, and S. Sarkar, “Throughput guarantees in maximal scheduling in wireless networks,” in *the 43rd Annual Allerton Conference on Communication, Control and Computing*, Sep. 2005.
- [7] X. Wu and R. Srikant, “Scheduling Efficiency of Distributed Greedy Scheduling Algorithms in Wireless Networks,” in *IEEE INFOCOM 2006*, Barcelona, Spain, Apr. 2006.
- [8] J. Ni and R. Srikant, “Distributed CSMA/CA Algorithms for Achieving Maximum Throughput in Wireless Networks,” in *Information Theory and Applications Workshop*, Feb. 2009.
- [9] A. Dimakis and J. Walrand, “Sufficient Conditions for Stability of Longest-Queue-First Scheduling: Second-Order Properties Using Fluid Limits,” *Advances in Applied Probability*, vol. 38, no. 2, pp. 505-521, 2006.
- [10] C. Joo, X. Lin, and N. Shroff, “Understanding the Capacity Region of the Greedy Maximal Scheduling Algorithm in Multi-Hop Wireless Networks,” in *IEEE INFOCOM 2008*, Phoenix, Arizona, Apr. 2008.
- [11] G. Zussman, A. Brzezinski, and E. Modiano, “Multihop Local Pooling for Distributed Throughput Maximization in Wireless Networks,” in *IEEE INFOCOM 2008*, Phoenix, Arizona, Apr. 2008.
- [12] M. Leconte, J. Ni, and R. Srikant, “Improved Bounds on the Throughput Efficiency of Greedy Maximal Scheduling in Wireless Networks,” in *ACM MOBIHOC*, May 2009.
- [13] A. Proutiere, Y. Yi, and M. Chiang, “Throughput of Random Access without Message Passing,” in *Conference on Information Sciences and Systems*, Princeton, NJ, USA, Mar. 2008.
- [14] R. R. Boorstyn, A. Kershenbaum, B. Maglaris, and V. Sahin, “Throughput Analysis in Multihop CSMA Packet Radio Networks,” *IEEE Transactions on Communications*, vol. 35, no. 3, pp. 267-274, Mar. 1987.
- [15] G. Bianchi, “Performance Analysis of the IEEE 802.11 Distributed Coordination Function,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535-547, 2000.

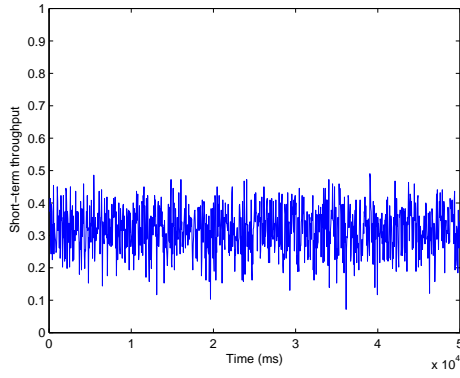
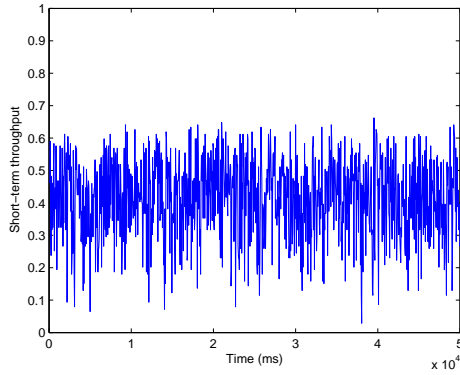
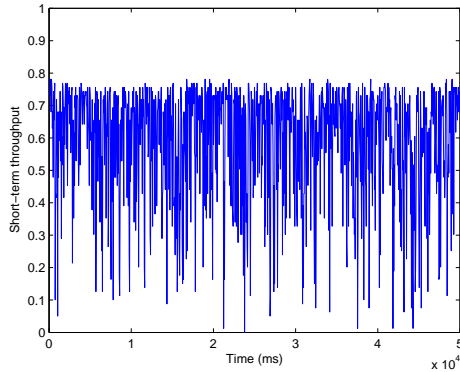
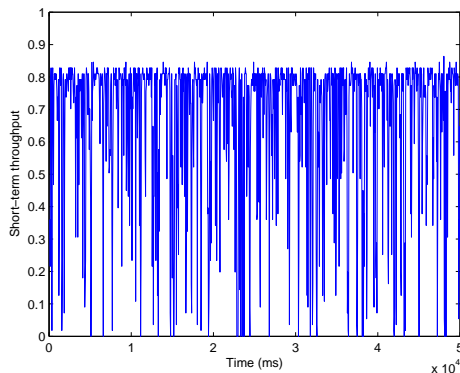
(a) $T^P = 25$ (b) $T^P = 40$ (c) $T^P = 70$ (d) $T^P = 100$

Fig. 14: Short-term throughput of link 13 in the 2-D lattice network

- [16] L. Jiang and J. Walrand, "A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks," in the *46th Annual Allerton Conference on Communication, Control, and Computing*, Sep. 23-26, 2008.
- [17] S. Rajagopalan and D. Shah, "Distributed Algorithm and Reversible Network", in *Conference on Information Sciences and Systems*, Princeton, NJ, USA, Mar. 2008.
- [18] P. Marbach, A. Eryilmaz, and A. Ozdaglar, "Achievable Rate Region of CSMA Schedulers in Wireless Networks with Primary Interference Constraints," in *IEEE Conference on Decision and Control*, 2007.
- [19] L. Jiang and S. C. Liew, "Improving Throughput and Fairness by Reducing Exposed and Hidden Nodes in 802.11 Networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 34-49, Jan. 2008.
- [20] C. Bordenave, D. McDonald, A. Proutiere, "Performance of random medium access control, an asymptotic approach," in *Proceedings of the ACM Sigmetrics 2008*, pp. 1-12.
- [21] F. P. Kelly, "Loss networks," *Ann. Appl. Prob.*, vol. 1, no. 3, 1991.
- [22] J. Liu, Y. Yi, A. Proutiere, M. Chiang, and H. V. Poor, "Towards utility optimal random access without message passing," (Invited Paper) *Wiley Journal on Wireless Communications and Mobile Computing, Special Issue on Advances in Wireless Communications and Networking*, vol. 10, no. 1, pp. 115-128, Jan. 2010.
- [23] V. Borkar, "Stochastic Approximation: A Dynamical Systems Viewpoint," 2008.
- [24] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, "New insights from a fixed point analysis of single cell IEEE 802.11 WLANs," in *Proc. IEEE INFOCOM*, USA, Mar. 2005.
- [25] X. Wang and K. Kar, "Throughput Modelling and Fairness Issues in CSMA/CA Based Ad-Hoc Networks," in *IEEE INFOCOM 2005*, Miami, Florida, Mar. 2005.
- [26] M. Durvy, O. Dousse and P. Thiran, "On the Fairness of CSMA Networks," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 7, pp. 1093-1104, Sep. 2009.
- [27] S. C. Liew, C. Kai, J. Leung, and B. Wong, "Back-of-the-Envelope Computation of Throughput Distributions in CSMA Wireless Networks," in *IEEE ICC*, 2009.
- [28] P.M. van de Ven, J.S.H. van Leeuwen, T.J.J. Denteneer, A.J.E.M. Janssen, "Spatial fairness in wireless multi-access networks," in *Proc. 4th International Conference on Performance Evaluation Methodologies and Tools (ValueTools'09)*, Pisa, Italy, Oct. 2009.
- [29] L. Jiang and J. Walrand, "Approaching Throughput-Optimality in a Distributed CSMA Algorithm: Collisions and Stability," (invited), *ACM Mobihoc'09 S3 Workshop*, May 2009.
- [30] L. Jiang and J. Walrand, "Approaching throughput-optimality in a Distributed CSMA Algorithm with Contention Resolution," Technical Report, UC Berkeley, Mar. 2009. URL: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-37.html>
- [31] L. Jiang, D. Shah, J. Shin, J. Walrand, "Distributed Random Access Algorithm: Scheduling and Congestion Control," <http://arxiv.org/abs/0907.1266>
- [32] L. Jiang and J. Walrand, "A Distributed CSMA Algorithm for Throughput and Utility Maximization in Wireless Networks," Technical Report, UC Berkeley, Aug. 2009. <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-124.html>

PLACE
PHOTO
HERE

Libin Jiang received his Ph.D. degree in Electrical Engineering & Computer Sciences from the University of California at Berkeley in 2009. Earlier, he received the B.Eng. degree from the University of Science and Technology of China in 2003, and the M.Phil. degree from the Chinese University of Hong Kong in 2005. His research interests include wireless networks, communications and game theory. He received the David J. Sakris Memorial Prize in 2010 for outstanding doctoral research, and the best presentation award in the ACM Mobihoc'09

S3 Workshop.



PLACE
PHOTO
HERE

Jean Walrand received his Ph.D. in EECS from UC Berkeley, where he has been a professor since 1982. He is the author of *An Introduction to Queueing Networks* (Prentice Hall, 1988) and of *Communication Networks: A First Course* (2nd ed. McGraw-Hill, 1998) and co-author of *High Performance Communication Networks* (2nd ed, Morgan Kaufman, 2000). Prof. Walrand is a Fellow of the Belgian American Education Foundation and of the IEEE and a recipient of the Lanchester Prize and of the Stephen O. Rice Prize.